Mendelova univerzita v Brně
Provozně ekonomická fakulta

Optimalizace antiplagiátorského řešení na Mendelově univerzitě v Brně

Diplomová práce

Vedoucí práce:
Mgr. Tomáš Foltýnek, Ph.D.

Jan Floryček

Brno 2015

Rád bych poděkoval vedoucímu mé práce, Mgr. Tomáši Foltýnkovi, Ph.D., za vstřícný přístup, odborné vedení, mnoho podnětných návrhů a cenných připomínek.

Čestné prohlášení

Prohlašuji, že jsem práci Optimalizace antiplagiátorského řešení na Mendelově univerzitě v Brně vypracoval samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a v souladu s platnou Směrnicí o zveřejňování vysokoškolských závěrečných prací.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 2. ledna 2015

_____

podpis

Abstract

The aim of this thesis is to analyse and optimise the efficiency of the anti-plagiarism system Anton that is being used at Mendel University in Brno. In this thesis I deal with evaluating of the current approach from the performance and accuracy points of view. A number of tests were prepared and executed, then the results are discussed. An option to cheat the system is also mentioned and tested.

Keywords

Plagiarism, Anton, optimization, PHP

Abstrakt

Cílem této práce je analyzovat a optimalizovat výkon antiplagiátorského systému Anton využívaného na Mendelově univerzitě v Brně. V této práci se zabývám hodnocením stávajícího řešení z hlediska výkonnosti a přesnosti. Byla připravena a realizována řada testů, následně jsou diskutovány výsledky. Rovněž je zmíněna a otestována možnost oklamání antiplagiátorského systému.

Klíčová slova

Plagiátorství, Anton, optimalizace, PHP

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod a cíl práce

1.1 Úvod

Plagiátorství je zejména v akademickém prostředí velice aktuální téma. Problematikou vydávání cizích děl za své se již zabývala celá řada jak teoretických, tak i praktických prací – z teoretických bych velmi rád zmínil například článek paní Němečkové (2008), obsáhlou teoretickou diplomovou práci Mgr. Černohlávkové (2008) či článek R. Harrise (2013). Z praktičtěji zaměřených prací dle mého názoru stojí za pozornost například test komerčních antiplagiátorských systémů od autorů Weber-Wulff, Möller, Touras, Zincke, (2013), dotazníkové šetření autorů Foltýnka, Rybičky a Demoliou (2013) či diplomová práce pana Veselého (2013). Přestože je plagiátorství fenomén existující již staletí, v poslední době lze vzhledem k rozvoji informačních technologií pozorovat výrazný nárůst zájmu o toto téma. Spolu se snadnou dostupností nejrůznějších materiálů narůstá i důležitost potírání plagiátorství ze strany vysokých škol a vědeckých komunit.

Návrh a implementace softwarových nástrojů pro odhalování plagiátů je tak dnes velmi perspektivní obor nejen pro fakulty se zaměřením na informační technologie, ale rovněž pro soukromé firmy, které mohou svá vlastní řešení nabízet všem zájemcům, zejména pak z oblasti školství.

Mendelova univerzita v Brně má k dispozici systém Anton, který dokáže odhalovat plagiáty mezi odevzdanými závěrečnými bakalářskými a diplomovými pracemi.

V této práci se nejprve budu zabývat plagiátorstvím z teoretického hlediska, budou vysvětleny základní pojmy a rozdíly mezi nimi, zmíněna související legislativa atd. Rovněž se budu zabývat různými přístupy a technikami boje proti plagiátorství – nejen softwarovými, ale bude zmíněna i problematika prevence.

Dále se budu zabývat efektivitou antiplagiátorského systému Anton – jeho rychlostí a přesností, vlivem nastavení na jeho funkčnost a případnými jeho slabinami a možnými vylepšeními do budoucna. S tím, jak se díky moderním technologiím rozšiřují možnosti samotného plagiátorství, je třeba stejně tak vylepšovat a optimalizovat i metody a technologie, jak jej potírat.

Zmíněna je i možnost záměrně oklamat antiplagiátorské systémy, což je velmi zajímavá oblast, kterou považuji z hlediska použitelnosti systému za velmi důležitou.

1.2 Cíl práce

Cílem práce je návrh na optimalizaci antiplagiátorského systomu Anton, který je k dispozici pro odhalování plagiátů na Mendelově Univerzitě. Systém bude analyzován, bude zkoumán výkon systému, jeho přesnost a vliv jednotlivých

parametrů na něj. Bude diskutováno, zda existuje prostor pro významnou optimalizaci systému.

Zmíněna bude také lingvistická stránka věci a možnosti záměrného oklamání antiplagiátorského systému.

2 Teoretický úvod do problematiky

V úvodu práce je třeba nejprve vysvětlit, co vlastně znamenají pojmy „plagiát“ a „plagiátorství“. Význam těchto slov je většině lidí intuitivně znám, nicméně přesná definice není zcela triviální z důvodů, které budou naznačeny dále v této kapitole. Samotný kořen slova pochází z latinského slova „plagiarius“, které znamená únos. V přeneseném smyslu slova se tak jedná o výraz označující únos cizích myšlenek.

Pokud jde o samotnou definici pojmů, rád bych uvedl například následující:

Česká terminologická databáze knihovnictví a informační vědy (TDKIV):
Plagiát: nedovolená napodobenina (přesná nebo částečná) uměleckého nebo vědeckého díla jiné osoby, která je bez uvedení předlohy vydávána za originál; její původce tak porušuje autorská práva autora původní předlohy.

Všeobecná encyklopedie DIDEROT:
Plagiát: Nedovolené napodobení, přejímání uměleckého, literárního nebo vědeckého díla bez udání vzoru nebo autora s cílem vydávat dílo za vlastní; dílo takto vzniklé.

Websterův encyklopedický slovník:
Využití slov nebo myšlenek jiného člověka, aniž by byly tomuto člověku přiznány zásluhy.

Norma ČSN ISO 5127-2003:
Představení duševního díla jiného autora půjčeného nebo napodobeného v celku nebo z části, jako svého vlastního.

Existuje celá řada dalších definic, nicméně všechny jsou si významově podobné. Je zřejmé, že u plagiátorství je klíčovým prvkem nedovolené napodobení či doslovné opsání cizí práce a vydávání takovéhoto díla za své, aniž by bylo uvedeno, že původním autorem je někdo jiný. Právě tímto se plagiát liší od kompilace. Definice kompilace zní dle TDKIV následovně:
Odborný text, který vznikl sestavením poznatků z jiných děl. Na rozdíl od plagiátu však nepřebírá celé hotové pasáže bez citování původního zdroje ani nepředstírá svou původnost.

2.1 Plagiátorství a česká legislativa

Česká legislativa sice přímo pojem plagiát nezná, to ovšem neznamená, že je otázka autorských práv a problematika vydávání cizích děl za svá opomíjena. Za zásadní lze v tomto ohledu považovat dva zákony:

- Zákon č. 121/2000 Sb. (Autorský zákon)
- Zákon č. 111/1998 Sb. (Zákon o vysokých školách)

Autorský zákon definuje autorské dílo jako „*...dílo slovesné vyjádřené řečí nebo písmem, dílo hudební, dílo dramatické a dílo hudebně dramatické, dílo choreografické a dílo pantomimické, dílo fotografické a dílo vyjádřené postupem podobným fotografii, dílo audiovizuální, jako je dílo kinematografické, dílo výtvarné, jako je dílo malířské, grafické a sochařské, dílo architektonické včetně díla urbanistického, dílo užitého umění a dílo kartografické.*“ Podle autorského zákona je je dílem například také počítačový program nebo databáze a rovněž dílo rozpracované nebo jeho část, případně tvůrčí zpracování jiného díla jako je překlad nebo kompilace.

Zároveň však zákon definuje také to, co dílem není a to následovně: *„Dílem podle tohoto zákona není zejména námět díla sám o sobě, denní zpráva nebo jiný údaj sám o sobě, myšlenka, postup, princip, metoda, objev, vědecká teorie, matematický a obdobný vzorec, statistický graf a podobný předmět sám o sobě.“*

Zákon o vysokých školách rovněž neřeší pojem plagiátorství jako takový. Obsahuje však povinnost zveřejňovat závěrečné práce v § 47b: *„Vysoká škola nevýdělečně zveřejňuje disertační, diplomové, bakalářské a rigorózní práce...“*. Tato povinnost na jedné straně umožňuje ověřování toho, zda práce nejsou opsané, na druhé straně však usnadňuje opisování samotné a vzhledem k množství prací znesnadňuje kontrolu. Za zmínku rovněž stojí, že český zákon o vysokých školách neřeší odebrání titulu v případě prokázání plagiátorství.

2.2 Citace

Pokud se zabýváme plagiátorstvím, je také třeba seznámit se s pojmem *citace*. Tento pojem definuje TDKIV jako „*Doslovné uvedení cizího výroku nebo textu v rámci vlastního dokumentu doprovázené obvykle přesnou identifikací pramene, ze kterého daný výrok nebo text pochází, tedy bibliografickou citací. Je typograficky odlišen od ostatního textu.*“ Citací se využívá například pro potřeby přesného definování pojmů, kdy je vhodné držet se jednoznačně zavedených standardů a parafráze tedy může být spíše na škodu.

Otázkami morálně správného využití výsledků cizí práce se v tomto kontextu věnuje citační etika. TDKIV ji definuje jako „Etické normy týkající se citování ve vědecké komunikaci. Všeobecně uznávaným principem citační etiky je morální povinnost autora publikované vědecké práce uvést v této práci ty výsledky svých myšlenkových předchůdců, na které ve své práci bezprostředně a vědomě navázal, a odlišit zřetelně své vlastní výsledky od výsledků jiných autorů. Za tímto účelem má autor práce těchto svých předchůdců citovat, zejména pak ty práce, jejichž výsledků využil (konstruktivní citace). Na druhé straně není správné spekulativně zatěžovat vědecké práce prestižními citacemi, jako jsou některé nekonstruktivní autoritativní citace a nekonstruktivní

autocitace.“ Citační etika se zabývá tím, jak správně citovat, kdy citovat a kdy naopak necitovat. Je například zbytečné, aby autor citoval například práci, kterou sice v rámci svého výzkumu četl, ale nijak konkrétně ji ve svém díle nevyužil. Naopak citovat by se měla každá práce, ze které autor cokoliv přebral, byť by to byla jen jediná definice, graf nebo obrázek. Správné citování všech využitých materiálů je nezbytné, pokud se chce autor vyhnout podezření z plagiátorství.

Souvisejícím termínem je *bibliografická citace*, jejíž definice je podle TDKIV „Formalizovaný údaj o dokumentu, který autor bezprostředně použil při přípravě své práce. Odkazy na citovaný dokument jsou v textu uvedeny formou číselných ukazatelů nebo zkrácených citací.“ Přesný obsah a forma bibliografických citací se v jednotlivých částech světa liší, pro Českou republiku platí normy ČSN ISO 690 (upravuje obsah, formu a strukturu) a ČSN ISO 690-2, která se zabývá citováním elektronických dokumentů nebo jejich částí. Normy připravila organizace ISO (Mezinárodní organizace pro normalizaci) – stanovují prvky citace, jejich pořadí, pravidla, formální úpravy atd. Každý autor odborných publikací by měl znát alespoň základní zásady těchto norem, aby mohl správně citovat.

Jak již bylo zmíněno výše, správným citováním všech zdrojů s dodržením patřičných norem se autor nemusí obávat, že by se dopustil plagiátorství – byť jen z omylu či nedbalosti. Vyhnutí se plagiátorství však v žádném případě není jediný důvod, proč by autoři měli citovat. Důvodů je mnoho a mezi nejdůležitější bych si dovolil zařadit následující:

- **Uznání zásluh** – citací cizího díla a uvedením původu si autor nepřivlastňuje cizí zásluhy. Naopak vyzdvihne tvůrčí činnost jiného odborníka.
- **Ověřitelnost uváděných informací** – uvedením zdroje informací (definice, letopočty, data…) autor umožní čtenáři, aby si ověřil pravdivost toho, co je mu předkládáno. Zároveň je možno zhodnotit důvěryhodnost uváděného zdroje.
- **Další zdroje informací** – uvedení zdroje lze považovat i za nasměrování čtenáře k dalším informačním zdrojům. Může tak snadno získat podrobnější informace o dané problematice a rozšířit své znalosti o oblasti, kterými se zabýval autor citovaného díla, ale autor nového díla je nevyužil.
- **Zasazení díla do kontextu** – uvedením autorů děl, ze kterých práce vychází, se čtenář může dozvědět, koho autor považuje za důvěryhodné odborníky. To je důležité zejména pro práce z oblastí, které nelze označit za exaktní vědu – například ekonomie, politologie nebo sociologie. Další důležitou informací může být například datum vydání citované práce – v celé řadě oblastí je vývoj natolik rychlý, že citovat díla starší než 10 let může být zbytečné nebo dokonce i vedoucí v omyl.

2.3 Důvody plagiátorství studentů

Příčin, proč se studenti dopouštějí plagiátorství je celá řada, na tomto místě uvedu několik z nich.

Důvody jsem zjišťoval od samotných studentů nejprve rozhovorem s několika desítkami současných a bývalých spolužáků a dále prostřednictvím krátkého online dotazníku, kde jsem získal odpovědi od dalších cca 130 respondentů z řad studentů. V dotaznících jsem žádal studenty, aby vybrali maximálně 10 možností z celkového počtu 20 nabízených (které byly vybrány na základě úvodních rozhovorů se samotnými studenty) na otázku, které důvody považují za nejpravděpodobnější důvody, proč se studenti uchylují k plagiátorství. Zde uvedené důvody jsou ty, které byly voleny nejčastěji. Některé z těchto důvodů a mnohé další zmiňuje ve své práci také Černohlávková (2008).

Považuji za velmi důležité tyto důvody znát, aby bylo možné s plagiátorstvím efektivně bojovat a předcházet mu.

Špatná organizace času

Jeden z nejčastěji udávaných důvodů je časová tíseň – s blížícím se termínem odevzdání práce se řada studentů v časové tísni uchyluje k plagiátorství, aby vůbec měli co odevzdat. Na tomto místě je však také třeba zmínit, že na mnoha školách se odevzdává několik rozsáhlých prací v termínech velmi blízko u sebe a tato ne právě ideální organizace může k plagiátorství studentů nepřímo přispět.

Nezajímavý nebo nedůležitý předmět či projekt

Studenti mají mnohdy pocit, že některé z předmětů, které v rámci svého oboru studují, příliš nesouvisí s jejich zaměřením. Takové předměty pak vnímají jako zbytečné a mají snahu je absolvovat s co nejmenší možnou námahou a s opisováním mají menší morální problém, než v předmětech, které jsou dle jejich názoru důležité a užitečné.

Absentující antiplagiátorská politika

Na řadě zejména menších a méně známých škol se doposud plagiátorství příliš neřeší. Neexistuje efektivní kontrola, jasná pravidla, poučení studentů nebo povědomí o trestech. A pokud neexistuje hrozba trestu, pak to jediné, co brání studentům v podvádění je vlastní svědomí.

Nedostatek vlastních schopností studenta

Mnoho studentů vysokých škol má jen velmi slabé schopnosti vyhledávat v knihovnách, databázích a katalozích a školy samotné aktivně tyto dovednosti zpravidla nevyučují. Zpravidla neexistují školení či kurzy, které by vyučovaly vyhledávání v informačních zdrojích.

Dalším problémem může být subjektivní náročnost některých předmětů či prací, která může v kombinaci s nedostatkem času vést k tomu, že student nebude k práci přistupovat poctivě.

Neosobní přístup
S tím, jak se zvyšuje počet vysokoškolských studentů – podle dokumentu Evropské komise (2010) Strategie Evropa 2020 by se mělo zvýšit procento vysokoškolsky vzdělaných lidí ve věku 30 až 34 let nejméně na 40 %, stává se čím dál náročnějším zachovat kvalitu studia, zejména co se týče přístupu vyučujících. Pokud kurz navštěvují desítky až stovky studentů, je v principu nemožné aby měli učitelé přehled o znalostech a schopnostech každého z nich. Následkem tohoto trendu je to, že se mnohdy hodnotí pouze odevzdané práce a ne studium jako celek. Mnohdy se práce dokonce hodnotí zcela automatizovaně (například v případě počítačových programů), aby se ušetřil čas. Negativa tohoto přístupu jsou zřejmá – studenti mnohdy ani neobhajují svou práci před člověkem, který ji hodnotí. Nedozví se, jakých chyb se dopustili a příště je udělají zase. Neprocvičí si schopnosti prezentovat výsledky své práce. A v neposlední řadě je tím usnadněno podvádění. Student totiž nemusí odevzdanou práci ani číst a vyučující si nevšimne, že se výrazně liší od předchozích prací stejného studenta.

Neschopnost rozlišit hranici mezi využití parafrázováním a plagiátorstvím
Při tvorbě odborných prací se velice často alespoň částečně vychází z již existujících prací na podobná témata. Pro nepoučeného studenta však může být problém rozlišit, co plagiátorství je a co není. Co se ještě považuje za všeobecnou znalost a co už je považováno za přebrání cizí myšlenky.

Neznalost citační etiky a norem či nedbalost
Mnozí studenti mají jen velice povrchní znalosti pravidel pro tvorbu citací. Často tak ani nemusí vědět, že se dopouští plagiátorství. Dalším důvodem může být nedbalost při psaní práce, kdy přebírají část cizího díla, nenapíší si zdroj, zpětně se jim už nepodaří nalézt a tak raději vydávají text za svůj, než aby jej z práce odstranili.

V této souvislosti stojí za zmínku kvalitní článek od autorů Foltýnka, Rybičky a Demoliou (2013). Jejich analýza vycházela z několika tisíc dotazníků po celé Evropě, osloveni byli studenti i učitelé.

Mezi 10 nejčastěji uváděnými důvody, proč se studenti uchylují k plagiátorství, skončily následující:

- Kopírování z internetu je snadné
- Plagiátorství není vnímáno jako velký prohřešek
- Předpoklad, že vyučujícímu to bude jedno
- Slabá schopnost porozumění textu
- Nepochopení rozdílu mezi skupinovou prací a nedovolenou spoluprací
- Nezbývá čas

- Neschopnost zvládnout množství práce
- Pocit, že vlastní práce není dost dobrá
- Pocit, že zadaná práce je nad jejich schopnosti
- Zadaná úloha je příliš náročná nebo nepochopená

Z výzkumu dále vyplynuly některé zajímavé výsledky, například:

- Pohled studentů a vyučujících na plagiátorství a postihy za něj se liší
- Studenti jsou o plagiátorství informování primárně z webu a ne prostřednictvím výuky
- Existuje nezanedbatelné procento univerzit, které studentům neposkytují žádný tréning v oblasti plagiátorství

2.4 Prevence plagiátorství

Plagiátorství je coby závažný prohřešek proti akademické etice a zpravidla také proti studijnímu řádu školy jev, se kterým je třeba bojovat. Vedle hrozby trestu či systémů pro usnadnění odhalování plagiátů je však také třeba se zabývat otázkou prevence, která by měla být součástí obecné antiplagiátorské politiky.

Problematiku prevence a obecného přístupu k plagiátorství velmi dobře popisují Mgr. Černohlávková (2008) či článek R. Harris (2013), z jejichž práce a vlastních zkušeností jsem v této podkapitole vycházel.

Je vhodné, aby komplexní antiplagiátorská politika byla jasně definována vedením školy – měla by zahrnovat prevenci, detekci plagiátů a postihy za prohřešky. Stejně tak by měla být jasně určená odpovědnost a kontaktní osoby. V rámci komplexního postupu proti plagiátorství bych na tomto místě rád uvedl některé postupy, které je možno považovat za preventivní opatření. Je totiž vhodné maximálně omezit příležitosti, motivovat studenty a poučit je o správné práci s informačními zdroji.

2.4.1 Poučení studentů

U studentů vysokých škol lze zpravidla očekávat nadprůměrnou inteligenci a schopnosti, nicméně by bylo chybou předpokládat, že budou mít všichni perfektní znalost problematiky plagiátorství. Naopak se lze domnívat, že na středních školách se práce se zdroji, citační normy nebo plagiátorství obecně řešilo jen velmi okrajově.

Jednou z možností je například kurz pro studenty, ve kterém se dozví vše potřebné – požadavky školy, normy, vysvětlení pojmů, důvody a motivace, poučení, seznámení s celkovou antiplagiátorskou politikou včetně možných následků a trestů. Je vhodné, aby byl kurz velmi prakticky zaměřen a studenti jej absolvovali již v prvním semestru a všechny informace tak získali co nejdříve během studia.

Další možnost je poučit studenty v rámci některých předmětů. Vyučující tak mohou zdůraznit to nejdůležitější z hlediska jejich konkrétního předmětu a

kdykoliv studentům připomenout pravidla. Nevýhodou naopak je zvýšená náročnost na koordinaci (aby byla pravidla jednotná a poučení se zbytečně neopakovala) a možný negativní vliv na výuku konkrétních předmětů.

Pomoci mohou samozřejmě i snadno dostupné pravidelně aktualizované materiály zpracované a schválené přímo školou, u nichž studenti nebudou muset pochybovat o jejich správnosti. Měly by být dostatečně stručné, aby se v nich dalo snadno orientovat, ale zároveň obsahovat všechny podstatné relevantní informace.

Všechny možné přístupy lze libovolně kombinovat pro co největší možnou efektivitu a dopad na studenty. Z mých diskuzí se studenty vyplývá, že v oblasti informování studentů má drtivá většina škol značné mezery. Žádný specializovaný kurz zpravidla neexistuje, informace o přístupu fakulty k plagiátorství se obtížně hledají a obsahují spíše obecné fráze a metody správného zacházení s informačními zdroji si studenti musí osvojit sami mimo výuku. Svým způsobem je to pochopitelné, bohužel tento přístup však vede k tomu, že se studenti dostávají k informacím pochybného původu a kvality a osvojují si tak špatné návyky.

Předmět poučení

Ať už by bylo poučení realizováno libovolným způsobem, studenti by se během něj měli osvojit celou řadu důležitých znalostí a zejména praktických dovedností, které budou moci využívat po celý zbytek studia i v případné následné akademické kariéře.

Začít lze například obecnou definicí pojmu plagiátorství. Vysvětlit studentům co to je plagiát a plagiátorství, popsat rozdíl mezi plagiátem, kompilací a parafrázováním. Zejména se zaměřit na plagiátorství v kontextu konkrétního studijního oboru, uvést vhodné příklady tak, aby se studenti naučili vnímat rozdíly mezi parafrází, kompilací a plagiátem.

Z ryze praktických dovedností by se studenti měli zejména naučit, jak správně citovat a že je třeba vždy řádně citovat všechny použité zdroje, i v případě že neopisují text doslovně, ale parafrázují jej. Vysvětlit pojmy jako citační etika, citační normy, rozdíly mezi citací elektronických zdrojů, klasických knih a článků v odborných časopisech. V souvislosti s citováním by také bylo vhodné zdůraznit, že ne všechny zdroje, zejména pak veřejně dostupné elektronické zdroje, u kterých nelze dohledat původ, jsou vhodné pro citování v odborných pracích. Citování zdánlivě obsáhlého webu může být lákavé pro rychlost a dostupnost, nicméně takovýto přístup může snadno snížit celkovou kvalitu práce, pokud si student nedá práci a informace si neověří.

Další důležitou informací, kterou by si poučení studenti měli odnést je, že škola se řešením plagiátorství aktivně zabývá a existuje možnost odhalení a následného trestu. Již samotná hrozba postihu může vést ke sníženému procentu studentů, kteří se k plagiátorství uchýlí. Studenti by měli být informováni například o tom, zda škola využívá nějaký elektronický nástroj pro snadnější odhalování plagiátů a popsat postup jak se podezření na plagiátorství řeší, případné tresty a podobně.

2.4.2 Způsob zadávání a hodnocení prací

Vedle poučených studentů má na plagiátorství vliv rovněž způsob, jakým se předměty vyučují, zejména pak zadávání a hodnocení prací. Studenti by měli mít motivaci se snažit podávat co nejlepší výkon a jeden ze způsobů jak toho dosáhnout je zadávání smysluplných prací. Zadaná práce by měla být nějakým způsobem zajímavá a měl by v ní být prostor pro kreativní přístup. Mnozí ze studentů, se kterými jsem na toto téma mluvil, si stěžovali na to, že témata prací se neustále opakují a jsou nepříliš zajímavá. Často je pak navíc student za takovouto práci hodnocen pouze na nějaké formě číselné stupnice a už se nedozví, jaké konkrétní aspekty byly zpracovány lépe a jaké hůře. Toto všechno značně demotivuje studenty, jejichž případná snaha není řádně doceněna.

Uvedené problémy lze alespoň částečně eliminovat změnou přístupu – například zadání prací vždy nějakým způsobem pozměnit, aby se omezila možnost nedovolené spolupráce mezi studenty, případně opisování starších prací na stejné téma. Další možností je umožnit studentům přijít s vlastním návrhem tématu, které je jim blízké a u kterého by mohli předvést vlastní iniciativu a kreativní přístup – a ne pouze psát práce na témata, která již byla mnohokrát zpracovaná.

S tím souvisí i přístup vyučujících, kteří by v ideálním případě měli mít i alespoň základní přehled o schopnostech svých studentů, aby poznali případnou změnu odborné nebo stylistické úrovně práce oproti předchozím pracím. Zejména by však práce měla být komplexně ohodnocena, aby se student dozvěděl, jaké měla nedostatky a jakých chyb by se měl příště vyvarovat. Tímto lze očekávat zvýšení studentových schopností do budoucna a přispět k jeho motivaci. Hodnocení práce je navíc možno spojit s nějakou formou obhajoby, kde si student osvojí schopnost prezentovat své dílo před další osobou a navíc může posloužit jako další forma prevence plagiátorství – obhajovat opsanou práci je náročnější, než obhájit práci kterou student zpracoval poctivě. Zpětná vazba je pro studenta celkově prospěšná a proto by jí měla být věnována maximální pozornost.

V neposlední řadě je také třeba zmínit otázku organizace času a rozvržení práce během semestru. Jak již bylo uvedeno výše, existují studenti, kteří se k plagiátorství uchylují mimo jiné kvůli nedostatku času. Přestože není v silách samotné školy tento problém zcela eliminovat, lze mu alespoň částečně předcházet tím, že budou vyučující jednotlivých předmětů spolupracovat při zadávání prací a termínů odevzdávání. V ideálním případě by zadání měla být zveřejněna již na začátku semestru a termíny odevzdání by neměly být příliš kumulovány do jednoho období.

2.5 Co je a co není plagiátorství

V předchozích kapitolách bylo uvedeno několik pojmů, které s plagiátorstvím souvisí. Na tomto místě bych je rád rozebral podrobněji a zasadil je do kontextu. Existuje řada názorů na to, co už je plagiátorství a co ještě není. Hranice mezi

plagiátem a kompilací či parafrázováním je velice tenká a pravděpodobně ji nelze vždy zcela jednoznačně určit. Jako jeden z kvalitních zdrojů však může posloužit článek Mgr. Němečkové z ČVUT, která se zde zabývala právě rozdíly mezi plagiátem a kompilací / parafrází a přinesla i doporučení, jak se plagiátorství, třeba i neúmyslnému, vyvarovat. Což je důležité nejen z hlediska akademické etiky a morálních zásad, ale také kvůli možným následkům. Za zmínku stojí i fakt, že plagiátorství se může dopustit nejen osoba, která cizí dílo neoprávněně využije ve svůj prospěch, ale i ten, kdo takovéto jednání umožňuje nebo k němu nabádá.

2.5.1 Co nelze označit za plagiátorství

Plagiátorstvím rozhodně není veškeré využívání cizích děl – naopak, mnoho odborných prací využívá alespoň částečně starších prací. Může se jednat o využití teoretických pasáží vysvětlující úvod do popisované problematiky nebo o zapracování cizích závěrů, na které může nová práce navázat a rozšířit tak poznání o zkoumané oblasti. Proto je důležité uvést některé příklady využití cizích děl, které jsou z etického i právního hlediska v pořádku a nejedná se o plagiátorství jako takové.

Kompilace
Kompilace je dílo, které vzniklo kombinací děl již dříve existujících. V kontextu odborných textů je kompilace typicky dílem, které spojí poznatky z několika starších děl na podobné téma do nového celku, který může nabídnout například propojení více úhlů pohledu nebo obsáhlejší rozbor určité problematiky. Kompilace nemusí obsahovat žádný nový tvůrčí poznatek autora ani výsledek výzkumné činnosti. Podmínkou ovšem je, aby byly všechny zdroje řádně citované a celá práce byla prezentována jako kompilát děl starších. Pokud to není nezbytně nutné, měl by se autor kompilace vyhnout doslovnému kopírování celých rozsáhlých pasáží a přílišnému využívání pouze jednoho zdroje.

Parafráze
Parafráze je vyjádření původního obsahu jiným způsobem, v případě psaní textu to znamená použití vlastních slov k popsání myšlenek, které nepřinesl sám autor. Jiná formulace základních myšlenek, se kterými jako první přišel někdo jiný, není plagiátorství, ovšem samozřejmě pouze za předpokladu, že bude původní dílo řádně uvedeno a citováno.

Všeobecně známá fakta
Všeobecně známá fakta není v odborných textech třeba citovat. Za všeobecně známá fakta se v tomto kontextu považují nejen skutečnosti, které jsou známé široké veřejnosti, například významná historická data nebo běžně používané termíny, ale rovněž základní terminologii a znalosti v daném oboru. Jedná se o skutečnosti a termíny, jejichž znalost lze u cílové skupiny čtenářů očekávat a

které lze v případě potřeby bez problémů dohledat v učebnicích či encyklopediích. Za tyto základní znalosti lze považovat například vzorce chemických sloučenin, základní matematická pravidla a další všeobecně známé vědecké poznatky. Existují však situace, kdy může být obtížné určit, zda se jedná ještě o všeobecně známou skutečnost, nebo zda je již popisovaná problematika natolik specializovaná a neznámá, že by měl zdroj být citován. V případě pochybností lze doporučit raději citovat, popřípadě se poradit například s vedoucím práce či jiným znalcem v daném oboru.

2.5.2 Co lze označit za plagiátorství

Za plagiátorství lze označit celou řadu přístupů a jednání, nejen úmyslné doslovné zkopírování cizích děl. V této podkapitole budou rozebrány některé z nich. Přestože se od sebe jednotlivé typy plagiátorství liší, všechny patří mezi nevhodné, pokud jde o psaní odborných prací. Zároveň je však třeba si uvědomit, že různé druhy plagiátorství mohou být různě závažnými prohřešky – v závislosti na okolnostech.

Úmyslný plagiát
Úmyslný plagiát je zřejmě nejčistším příkladem plagiátorství – vzniká tehdy, když se někdo úmyslně dopustí opsání či kopírování cizího díla a vydává jej za vlastní výtvor, aniž by uvedl autora originálu a správně citoval. Cílem takového jednání bývá usnadnit si práci a získat neoprávněnou výhodu.

O plagiátorství se jedná nejen v případě opsání kompletního textu, ale i tehdy, když je převzata pouze část – například vybrané pasáže, obrázky, tabulky či grafy. Autor originálu a původní zdroj se neuvádí z důvodu, aby se co nejvíce zkomplikovalo případné dohledávání a prokazování. Rovněž je důležité si uvědomit, že ani volně dostupné dílo nebo dílo, k jehož autorství se žádná konkrétní osoba nehlásí, si nelze přivlastnit a vydávat jej za své.

Nedodržení citační etiky
Citování by se mělo řídit platnými normami a dodržovat všeobecně uznávané zásady. Pokud jsou citace nedostatečné či nepřesné, lze hovořit o plagiátorství – například v případě doslovné citace cizího textu, aniž by byla pasáž takto označena.

Nesprávná kompilace
Jak bylo uvedeno výše, kompilace je dílo vzniklé kombinací starších děl. I pro tvorbu kompilací však existují pravidla, jejichž porušením může vzniknout plagiát. Jedná se například o použití jednoho hlavního zdroje namísto zpracování několika různých zdrojů, nebo o doslovné opisování celých dlouhých pasáží. Takto vzniklá díla lze označit za plagiát.

Nesprávná parafráze
O nesprávnou parafrázi se jedná tehdy, když autor nepřejímá pouze základní myšlenky, ale přepisuje celé pasáže textu vlastními slovy, aniž by uvedl zdroj a řádně citoval. V takovém případě se nemusí jednat o správně provedené parafrázování, ale může se již jednat o plagiátorství – byť z nedbalosti.

Nesprávné rozpoznání všeobecně známých faktů
Chybou může být i nesprávný odhad toho, co je a co není všeobecně známý fakt. Správné rozlišení může být o to náročnější, že tvůrce odborné práce četl mnoho zdrojů na dané téma, čímž může být značně zkreslen jeho odhad. Jak bylo zmíněno výše, při pochybách je vhodné se poradit s odborníkem v oboru z důvodu vyhnutí se možným pozdějším nepříjemnostem a podezření z plagiátorství.

Využití vlastních děl
Nedovoleného využití vlastního díla neboli auto-plagiátorství se dopustí autor, který v nové práci využije texty ze své vlastní starší práce a neocituje je. I toto chování je nepřípustné, byť podstatou odlišné od klasických typů plagiátorství.

Kryptomnézie
Neboli skrytá paměť je případ neúmyslného plagiátorství, když autor práce využívá myšlenku, jejíž zdroj si již nepamatuje a není schopen jej dohledat. Může se stát i to, že myšlenku, o níž někde četl či slyšel, vydává omylem za svou, aniž by vědomě a záměrně lhal.

2.6 Rozpoznávání plagiátů

Rozpoznávání plagiátů je vzhledem k množství odevzdávaných prací a zejména množství dostupných zdrojů velmi náročný a zdlouhavý úkol. Obvykle se k němu využívá softwarových nástrojů, nicméně existuje mnoho indicií, které mohou naznačovat, že student nevypracoval svou práci zcela sám a poctivě. Jejich přítomnost však samozřejmě automaticky neznamená, že práce je plagiát a naopak jejich nepřítomnost nemusí nutně znamenat, že práce plagiát není. V případě textových dokumentů se jedná například o následující skutečnosti:

- **Nejednotné formátování dokumentu** – pokud se v dokumentu bezdůvodně vyskytují různé fonty, různé velikosti písma, styly odstavců a další případy rozdílného formátování, je možné, že texty byly beze změny zkopírovány z různých zdrojů.
- **Nejednotný literární styl dokumentu** – lze očekávat, že práce psaná jedním člověkem bude konzistentní i po obsahové stránce. Pokud se v textu vyskytují kvalitativně výrazně rozdílné pasáže, chyby vyskytující se jen v některých částech nebo si dokonce jednotlivé části textu nějakým způsobem odporují, lze to chápat jako znak opsaného textu.

- **Odkazy na neexistující části** – občas studenti kopírují části jiných dokumentů beze změny a nevšimnou si, že obsahují například odkazy na obrázky, tabulky nebo poznámky pod čarou. Pokud chybí odkazovaný objekt, může to znamenat, že text byl do práce zkopírován odjinud.
- **Nedbalé citování** – zdroje nejsou uvedeny vůbec, je záměrně komplikována jejich identifikovatelnost, jsou uváděny zdroje, které v práci nebyly použity atd. Všechno toto naznačuje, že si autor nepřeje, aby někdo zkoumal skutečné zdroje, ze kterých vycházel.

Popsané indicie nemusí samozřejmě nic znamenat, a pokud je podvádějící student pozorný, dá si na ně pozor.

2.7 Automatizované antiplagiátorské systémy

S rozvojem informačních technologií došlo nejen k usnadnění podvádění (webové stránky poskytující práce na různá témata, online knihy, snadný přístup do katalogů knihoven a databází…), ale rovněž se rozšířily možnosti, jak se mohou vysoké školy proti plagiátorství bránit. V této kapitole bych rád popsal, jak antiplagiátorské systémy obecně fungují a uvedl několik příkladů existujících řešení.

Obecně lze říci, že automatizované nástroje pro odhalování plagiátů mohou být velice užitečné a výrazně usnadní učitelům práci. Při počtu studentů, kterých bývají v závislosti na studijním oboru desítky až stovky v ročníku a množství zdrojů které jsou k dispozici by bylo v podstatě nemožné, aby byly práce kontrolované na plagiátorství bez použití softwaru. Je však třeba si uvědomit že i ten nejlepší software bude mít stále jistá omezení, která nejspíše nepůjde nikdy zcela eliminovat. Vedle pořizovacích a případně provozních nákladů (do kterých patří nejen pořízení samotného softwaru, ale i nutnost školení zaměstnanců v jeho používání a pochopitelně čas, který budou jeho obsluhou muset strávit) stojí za zmínku stojí například následující:

- **Problém vstupních dat** – v knihovnách, na internetu a v nejrůznějších neveřejných databázích je takové množství dat (a neustále přibývají další), že není možné, aby byly zpracovány a s novou prací porovnávány všechny. Už toto je principiální problém, díky němuž nelze nikdy s naprostou jistotou říci, že odevzdaná práce není plagiát.
- **Problém výpočetního výkonu** – existuje mnoho metod a přístupů, jak porovnávat text. Některé jsou efektivnější a rychlejší než jiné, stále však existuje základní problém, že s rostoucím množstvím dat, která je třeba zpracovat, poroste i časová náročnost tohoto zpracování.
- **Problém FAR / FRR** – nesprávná akceptace (FAR = False Acceptance Rate) a nesprávné odmítnutí (FRR = False Rejection Rate) jsou pojmy známé z oblasti biometrických systémů. Vyskytují se ale ve všech oblastech, kde se srovnává podobnost dvou nebo více položek. Pokud je

systém nastaven na požadavek příliš vysoké shody, bude často docházet k tomu, že vzorky, které by měly být označeny za shodné, takto označeny nebudou. Naopak pokud bude práh shody nastaven příliš nízko, budou za shodné označovány vzorky, které shodné nejsou. Ať je však nastavení jakékoliv, docházet bude občas k oběma situacím.
- **Nutnost lidské kontroly** – díky výše popsaným problémům je nutné, aby výstupy systému kontroloval člověk. To s sebou však bohužel nese další požadavky na něčí čas.

I přes výše zmíněné nedostatky, jejichž výčet není v žádném případě kompletní, je však třeba si uvědomit, že automatizované detekční nástroje umožňují míru kontroly, která by bez jejich použití byla zcela nemyslitelná. Navíc už samotné povědomí o tom, že škola s plagiátorstvím aktivně bojuje s využitím specializovaných systémů, může mít na řadu potenciálních plagiátorů preventivní efekt.

Co se týče konkrétních postupů zpracování a metod detekce, existují dva typy systémů, které jsou zajímavé z hlediska kontroly prací na vysoké škole technického typu:
- Systémy pro detekci plagiátorství v textových dokumentech
- Systémy pro detekci plagiátorství ve zdrojových kódech počítačových programů

Oba typy sdílí některé základní myšlenky a metody, nicméně jejich povaha je natolik odlišná, že se jejich stručnému popisu budu věnovat samostatně.

2.7.1 Systémy pro detekci plagiátorství ve zdrojových kódech

Detekovat plagiátorství ve zdrojových kódech programů může být ještě obtížnější úkol, než v případě textových dokumentů. Důvodem je fakt, že pokud plagiátor vyvine alespoň minimální úsilí, lze zdrojový kód na první pohled poměrně výrazným způsobem změnit od originálu, aniž by se přímo změnila jeho funkčnost. Některé úpravy zmiňuje například Clough (2000) ve své práci:
- Změna komentářů
- Změna datových typů
- Změna názvů identifikátorů
- Změna názvů funkcí, tříd a metod
- Změna pořadí příkazů či funkcí
- Přidání nadbytečných příkazů a proměnných
- Změna struktury kódu, nahrazení příkazů ekvivalentními (cykly, logické podmínky…)
- Přepsání kódu do jiného programovacího jazyka
- Rozdělení jednoho zdrojového kódu do několika samostatných souborů

Část problémů lze eliminovat například striktnější podobou zadání – přesně určit programovací jazyk, třídy, funkce, metody atd. Následek by však bylo omezení kreativity studentů, tudíž tento přístup nelze příliš doporučit. Z výše uvedeného ovšem vyplývá, že prosté porovnávání textu je minimální, avšak nikoliv postačující způsob jak detekovat plagiátorství ve zdrojových kódech – už při relativně jednoduchých změnách lze docílit výrazně odlišného zápisu kódu. Existuje nepřeberné množství způsobů, jak refaktorovat zdrojový kód a proto je nezbytné, aby tuto skutečnost nástroje pro práci s ním braly v potaz.

Oblast detekce plagiátorství ve zdrojových kódech je velice rozsáhlá a zajímavá, za velmi kvalitní práci zabývající se touto tématikou považuji text pánů Chanchala a Cordyho (2007). V této práci se autoři zabývají mimo jiné klasifikací detekčních technik – každá z nich má své specifické vlastnosti, na jejichž základě ji lze zařadit. Mezi zmíněné vlastnosti patří například následující:

Transformace / Normalizace zdrojového kódu
Jak již bylo naznačeno výše, pracovat se samotným zdrojovým kódem bez jakýchkoliv úprav není příliš vhodné. Využívají se tedy přístupy, které před samotným porovnáním zdrojový text upraví a normalizují. Některé jen odstraní / upraví bílé znaky (například jejich sekvence nahradí jen jedním znakem) a komentáře, jiné mohou aplikovat komplexnější transformace a více pozměnit reprezentaci kódu (například úpravy názvů proměnných, nahrazení příkazů ekvivalentním a podobně). Výsledná forma by měla být vhodnější pro porovnání a vyhledávání podobností.

Reprezentace zdrojového kódu
Vhodná reprezentace zdrojového kódu je získána aplikací transformačních a normalizačních technik, aby co nejlépe vyhovovala danému porovnávacímu algoritmu.

Granularita porovnávání
Různé porovnávací algoritmy pracují s různě velkými elementy. Některé pracují s příkazy, jiné s tokeny a další mohou pracovat například s uzly abstraktních syntaktických stromů.

Porovnávací algoritmus
Existuje mnoho porovnávacích algoritmů a volba toho správného pro daný účel je klíčová. Pro hledání podobností lze využít algoritmy z různých oblastí – některé přístupy využívají algoritmy shody sekvencí (sequence matching algorithms), které se běžně využívají v bioinformatice pro porovnávání biologických dat – například řetězců DNA. Jiné přístupy mohou využívat algoritmy z oblasti data miningu – například neuronové sítě.

Výpočetní náročnost
Výpočetní náročnost algoritmu je velmi důležitá vlastnost, protože je třeba vzít v potaz nejen množství řádků kódu (ve školních pracích řádově zcela běžně stovky

až tisíce), ale rovněž počet zdrojových kódů, které je třeba vzájemně porovnávat. Náročnost je dána jednak typem a počtem transformací a pak také použitým porovnávacím algoritmem.

Podobnost klonů
Tato vlastnost udává, jaké typy klonů lze danou metodou detekovat. Některé techniky mohou najít pouze přesnou shodu, jiné mohou najít přibližnou shodu na základě zvolených parametrů.

Jazyková nezávislost
Velmi významným faktorem, který je třeba mít na paměti je otázka programovacího jazyka. Programovacích jazyků existuje celá řada – tato vlastnost udává, jaká jazyková podpora je potřebná pro zvolenou metodu.

Jazykové paradigma
Tato vlastnost udává, na jaké jazykové paradigma je cílena daná metoda. Některé metody mohou detekovat klony v procedurálních a objektově orientovaných jazycích, jiné v jazycích symbolických adres a podobně.

2.7.2 Proces detekce plagiátů ve zdrojových kódech

Existuje mnoho přístupů, jak vyhledávat plagiáty mezi zdrojovými kódy. Jak již bylo uvedeno, pro všechny fáze procesu existují různé algoritmy, které mají za úkol zdrojový kód nejprve transformovat a následně porovnávat jeho jednotlivé části. Obecný postup je velmi dobře popsán v textu Chanchala a Cordyho (2007).

Detekční nástroj má za úkol vyhledat části kódu, které jsou si ve srovnávaných souborech podobné. Vzhledem k možnému rozsahu kódu a faktu, že se musí porovnávat každá část zpracovávaného souboru s každou částí zdrojového souboru, se jedná o výpočetně velmi náročný proces. Z těchto důvodů se využívají postupy, jejichž úkolem je zmenšit doménu porovnávání ještě před tím, než se se samotným porovnáváním začne. Dalším problémem je, že nestačí pouze nalézt shodné fragmenty kódu – je zapotřebí další analýza, aby se detekovaly skutečné plagiáty. Na tomto místě budou popsány jednotlivé fáze procesu detekce. Je však třeba si uvědomit, že se jedná spíše o obecný popis – různé přístupy mohou mít různé fáze a využívat různé algoritmy pro jejich realizaci.

1. Předzpracování
Předzpracování bývá první fází detekčního procesu. Existují tři důležité cíle této fáze:

Odstranění nezajímavých částí: Veškerý zdrojový kód, který je nezajímavý pro porovnávání je v této fázi odstraněn. Například lze aplikovat partitioning na vložené části SQL kódu do kódu jiného programovacího jazyka, aby se různé

jazyky separovaly. Stejně tak je možné odstranit části kódu, u kterých je zvýšená šance na generování falešných shod (například inicializace tabulek).
Určení zdrojových jednotek: Po odstranění zdrojových jednotek je zbytek zdrojového kódu rozdělen do množiny disjunktních fragmentů, tzv. *zdrojových jednotek*. Tyto jednotky jsou největší části zdrojového kódu, které se budou přímo porovnávat. Existují různé úrovně granularity zdrojových jednotek – například třídy, funkce / metody, bloky begin-end, příkazy atd.
Určení porovnávaných jednotek / granularity: Zdrojové jednotky může být vhodné dále rozdělit na menší fragmenty, pokud je to vhodné pro zvolenou metodu komparace. Například je lze před porovnáváním rozdělit na jednotlivé řádky, nebo i tokeny. Rovněž lze využít syntaktické struktury jednotky - například blok příkazu *if* lze dále rozdělit na podmínku, větev *then* a větev *else*. Ne všechny metody porovnávání však vyžadují toto dělení – například metody založené na metrikách kódu, protože metriky lze vypočítat ze zdrojových jednotek libovolné granularity.

2. Transformace
Jednotky zdrojového kódu jsou v tomto kroku transformovány na jinou interní reprezentaci pro usnadnění porovnávání, nebo extrakci hodnot, které se budou porovnávat. Některé transformace jsou jednodušší – například odstranění bílých znaků a komentářů, jiné mohou být velmi komplexní – například generování grafu závislosti programu (PDG). V této části budou uvedeny některé transformační metody, které lze použít. Ne všechny metody se využívají pokaždé – vždy záleží na konkrétním přístupu. Níže jsou uvedeny některé z transformačních technik, které se aktuálně běžně využívají a upravují zdrojový kód předtím, než dojde k porovnávání:

Pretty printing zdrojového kódu: Jedná se o o metodu převodu zdrojového kódu do standardizované formy. Různé formátování je tak převedeno do jednotné podoby.
Odstranění komentářů: Drtivá většina metod je nastavena tak, aby ignorovala nebo odstranila komentáře ještě před samotným porovnáváním. Za zmínku stojí, že existují i přístupy, které komentáře využívají. Množství komentářů lze například použít jako jednu z metrik, případně lze porovnávat klíčová slova objevující se v komentářích.
Odstranění bílých znaků: Podobně jako u předchozího bodu, většina metod ignoruje / odstraňuje bílé znaky. I bílé znaky se však dají využít a počítat z nich například metriky layoutu.
Tokenizace: V metodách založených na tokenech je každý řádek zdrojového kódu rozdělen na tokeny podle lexikálních pravidel daného programovacího jazyka. Z těchto tokenů jsou následně utvořeny sekvence s tím, že veškeré bílé znaky a komentáře jsou odstraněny.
Parsování: Metody využívající parsovací stromy zpracují zdrojový kód a vytvoří z něj parsovací strom nebo abstraktní syntaktický strom (AST – Abstract Syntactic Tree). Následně lze porovnávat jednotky v podobě podstromů

parsovacího stromu nebo AST. Vedle přímého porovnávání podstromů a hledání plagiátů na základě podobnosti lze ještě počítat metriky těchto stromů a následně porovnávat podle nich.

Generování grafu závislosti programu: Existují přístupy využívající sémantická pravidla, které ze zdrojového kódu generují graf závislosti programu (PDG). Následně lze porovnávat jeho podgrafy – vyhledávat izomorfní podgrafy a na základě toho nalézat případné plagiáty. Samozřejmě lze také z grafu generovat metriky a provádět porovnávání na základě nich.

Normalizace identifikátorů: Většina metod provádí normalizaci identifikátorů a jejich nahrazení jediným tokenem před samotným porovnáváním.

Transformace elementů programu: Vedle normalizace identifikátorů lze na prvky zdrojového kódu aplikovat ještě další transformace. Různé varianty téhož syntaktického elementu (například cyklů) lze transformovat pro účinnější vyhledávání plagiátů.

Výpočet metrik: Jak již bylo zmíněno výše, existují metody založené na metrikách programu. Ty lze vypočítat jak přímo ze zdrojového kódu (počty řádků, identifikátorů, cyklů, komentáře…), tak z transformované reprezentace programu (parsovací stromy, AST, grafy a podobně).

3. Porovnávání shody

Do této fáze vstupuje text, který je již transformován – jednotlivé jednotky jsou vzájemně porovnávány a hledá se shoda. Využívá se i pořadí jednotek, ze kterých lze vytvořit větší shluky a porovnávat je. Obvykle se vyhledávají nejdelší společné sekvence porovnávaných textů.

Výstupem bývá seznam shodných pasáží v transformovaných kódech – páry jsou určeny informací o lokacích shodných fragmentů v transformovaném kódu. Například v metodách založených na tokenech je shodný pár reprezentován jako uspořádaná čtveřice (LeftBegin, LeftEnd, RightBegin, RightEnd) kde údaje reprezentují počáteční a koncovou pozici nalezené shodné sekvence v jednotlivých kódech.

Porovnávacích algoritmů lze v odborné literatuře najít velké množství – například Suffix Tree, Dynamic Pattern Matching nebo porovnávání hashe.

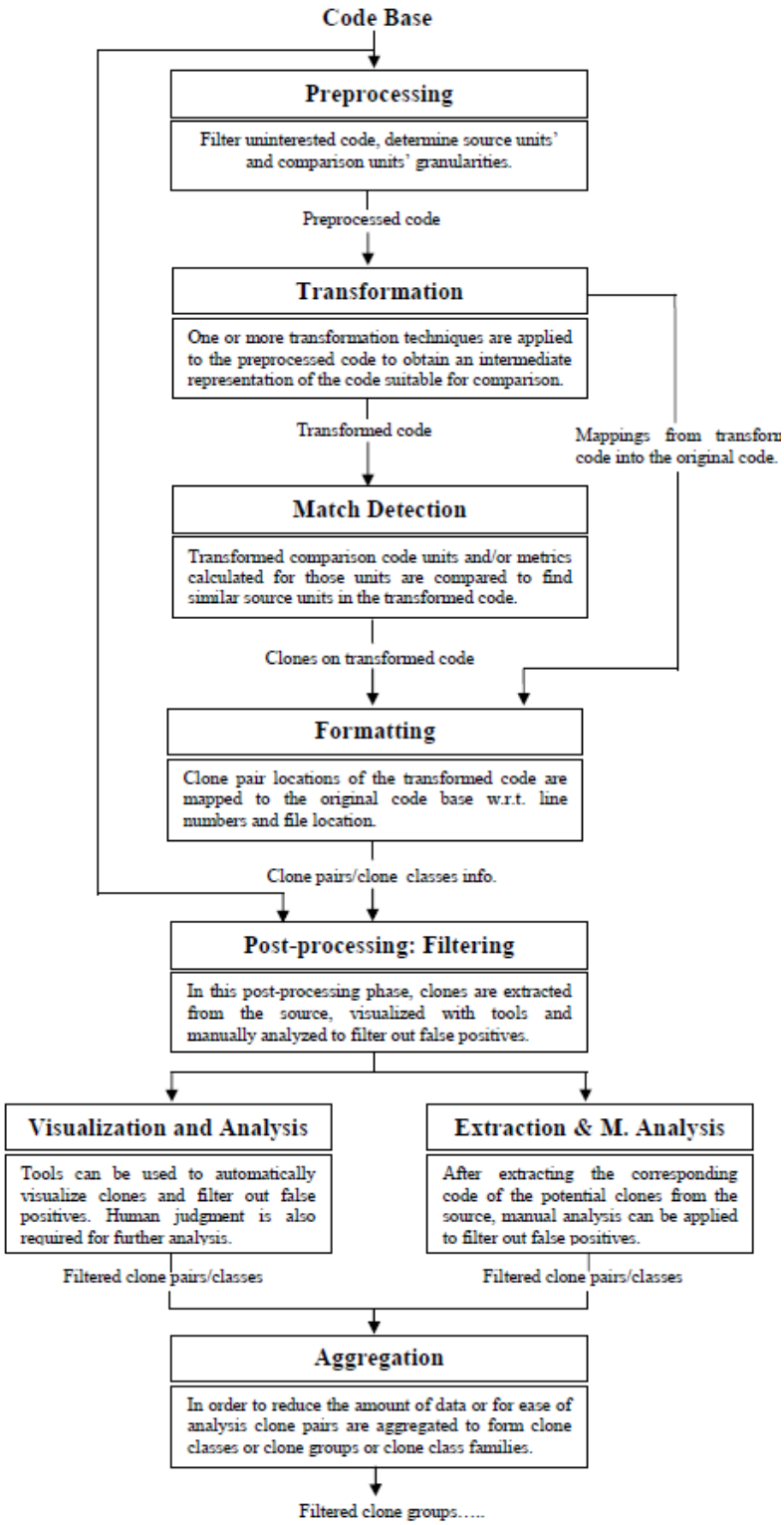
4. Formátování

Ve fázi formátování je třeba převést shodné fragmenty nalezené v transformovaném kódu na fragmenty v původní reprezentaci dat, tedy originálním zdrojovém kódu. Standardně jsou pozice shodných fragmentů nalezené v předchozí fázi převedeny na čísla řádků původního kódu.

5. Post Processing

V této fázi jsou odstraněny falešné shody. Je zde třeba manuální analýza člověka, zpravidla se využívá vizualizačních nástrojů pro zvýšení přehlednosti a zrychlení procesu.

Jak již bylo uvedeno výše – zde popsané fáze zpracování jsou velmi obecné a lze očekávat, že některé nástroje pro automatickou detekci plagiátů nevyužívají všechny fáze, nebo některé fáze realizují jinak, než je zde popsáno. Níže uvedené schéma zobrazuje návaznost jednotlivých fází procesu:



Obr. 1 Proces detekce plagiátorství ve zdrojových kódech, zdroj: Chanchal a Cordy (2007)

2.7.3 Klasifikace algoritmů pro porovnávání

Detekce plagiátů je proces sestávající primárně ze dvou fází – transformace a samotné porovnávání. Transformace má za úkol upravit vstupní kód tak, aby bylo následné porovnávání jednodušší a efektivnější. Následná porovnávací fáze pak detekuje shodné nebo podobné fragmenty. Jak bylo naznačeno výše, existuje mnoho algoritmů pro vyhledávání podobností ve zdrojových kódech. Využívají se různé přístupy, z nichž některé zde uvedu. Vycházel jsem z práce Chanchala a Cordyho (2007).

Podle ní existují následující kategorie klasifikace algoritmů pro vyhledávání podobností ve zdrojových kódech:

- Detekce založená na textu
- Detekce založená na tokenech
- Detekce založená na parsovacích stromech
- Detekce založená na grafech závislostí programu
- Detekce založená na metrikách
- Hybridní detekce

Detekce založená na textu
Tyto algoritmy vyhledávají přesnou shodu segmentů textů. Metody jsou obecně velmi rychlé, ale nepříliš efektivní, jelikož se dají oklamat prostým přejmenováním proměnných či názvů funkcí.

Detekce založená na tokenech
Detekce založená na tokenech funguje podobně jako detekce založená na textu, ale zdrojový kód je nejprve převeden na tokeny a tyto se následně využívají pro hledání shody. Převod na tokeny zajistí, že jsou ignorovány bílé znaky, komentáře a zejména názvy identifikátorů, čímž se systém stává robustnějším a odolnějším proti drobným změnám zdrojového kódu. Srovnávají se sekvence tokenů.

Detekce založená na parsovacích stromech
Tyto systémy vytváří a porovnávají parsovací stromy, čímž umožňují hledání podobností na vyšší úrovni. Například mohou normalizovat cykly nebo podmíněné příkazy a vyhodnocovat je jako podobné.

Detekce založená na grafech závislosti programu
Systémy založené na grafech využívají sémantické informace a informace o tocích řízení a tocích dat. Tím je umožněno srovnávání na ještě vyšší úrovni – vyhledávají se podobnosti pomocí algoritmu izomorfních podgrafů. Nevýhodou je vyšší výpočetní náročnost.

Detekce založená na metrikách
Existují systémy využívající metriky kódu – například počet cyklů, počet proměnných, počet podmínek atd. Tyto metody lze aplikovat relativně snadno,

ale nepovažují se za příliš spolehlivé – dva fragmenty kódu se stejnými metrikami mohou být naprosto rozdílné.

Hybridní detekce
Všechny výše popsané způsoby lze kombinovat – například nejprve využít méně výpočetně náročné metody a následně využít náročnější metody pouze na podmnožinu původních prací.

2.7.4 Systémy pro detekci plagiátorství v textových dokumentech

Podobně jako v případě detekce plagiátorství ve zdrojových souborech, i pro textové soubory existuje celá řada různých přístupů – lišících se výpočetní náročností i přesností.

Češka (2010) uvádí ve své práci následující klasifikaci algoritmů podle jejich komplexity:

Povrchní
Metriky jsou vypočítány bez jakékoliv znalosti lingvistických pravidel nebo struktury dokumentu.

Strukturální
Metriky jsou vypočítány s částečnou znalostí struktury dokumentu, například mohou být slova převedeny na jejich kořen nebo mohou být nahrazeny synonymy.

V textech pánů Steina (2008), Tetsuya, Kensuke, Yasuhiro a Daisuke (2011) a Řehůřka (2011) lze najít informace o mnoha konkrétních metodách, přičemž nejčastěji se jedná o následující:

Fingerprinting
Jedná se o metodu, při níž se vytvoří otisky (někdy také označován pojmy *hash* nebo *fingerprint*) podřetězců dokumentu, podle kterých je možno dokument identifikovat.

Metod jak určit vhodné podřetězce je mnoho a zpravidla berou v potaz atributy jako je pozice či frekvence výskytu. Mezi další faktory, které je třeba zvážit patří délka ukládaného hashe – příliš krátký hash znamená, že bude často docházet k nežádoucím kolizím. Příliš dlouhý hash bude mít naopak za následek zvýšené nároky ukládání dat i jejich zpracování. Důležitá je rovněž délka podřetězce, se kterým se pracuje – krátký podřetězec bude zbytečně generovat falešné shody, zatímco příliš dlouhý podřetězec bude příliš citlivý na drobné změny textu.

String Matching
V této metodě se vyhledává podřetězec, neboli *pattern* v delším textu. Lze využít substitucí a nehledat tak pouze konkrétní doslovný text, ale i podřetězce blízké zadanému.

Nevýhodou této metody je značná výpočetní náročnost a tedy nevhodnost pro rozsáhlé databáze dokumentů.

Vector Space Model
V tomto přístupu nejsou objekty reprezentovány přímo, ale jsou aproximovány tzv. *Features*. Každý objekt má přiřazenou hodnotu a výsledná reprezentace dokumentu je vektor těchto hodnot. Aproximace dokumentu sadou jeho jednotlivých prvků a jejich frekvencí se nazývá *Bag of Words*.

Existuje mnoho možných rozšíření a modifikací těchto nejběžnějších přístupů – lze například využívat neuronové sítě, sémantickou analýzu nebo stylometrickou analýzu. Tyto metody bývají však v odborné literatuře zmiňovány spíše okrajově.

2.8 Antiplagiátorský systém na Mendelově univerzitě v Brně

Mendelova univerzita v Brně má k dispozici antiplagiátorský systém zvaný Anton - The Antiplagiator Online Solution, jehož autory jsou Ing. Milan Šorm, Ph.D. a Mgr. Tomáš Foltýnek, Ph.D.

Systém je implementován v operačním systému Unix. Front end tvoří PHP kód využívající Nette framework a SQL. Podle autorů systému nabízí Anton stonásobné zrychlení oproti předchozí verzi implementované v UIS.

Základ algoritmu spočívá ve shlukování pětic slov, jejich seřazení v abecedním pořadí, vypočítání jejich MD5 hashe a uložení 8 bytů z tohoto hashe do databáze. Takto se projde celý dokument a postupně se uloží hashe všech jeho uspořádaných pětic. Množina hashů každého dokumentu, který je nově přidán do systému je porována s množinami hashů všech již existujících dokumentů a pokud je nalezena nenulová shoda, do databáze je přidán záznam o procentu této shody.

2.8.1 Postup zpracování dokumentů

Zpracování dokumentů systémem se skládá ze čtyř základních fází:

1. Dekódování
2. Předzpracování
3. Tvorba hashů
4. Porovnání

Každá z těchto fází připraví data pro následující fázi. Dokument musí projít všemi a finálním výstupem je procentuální shoda zpracovaného dokumentu s dokumenty, které byly zpracovány před ním.

1. Dekódování

V této úvodní fázi je dokument převeden z původního formátu na prostý text.

Podporované formáty jsou nejběžnější formáty dokumentů:

- PDF
- DOC
- DOCX
- HTML
- TXT

Nejsou podporovány v akademickém prostředí méně využívané formáty dokumentů jako ODT nebo EPUB, nicméně i ty lze v případě potřeby do podporovaných formátů snadno převést.

2. Předzpracování

Druhou fází je předzpracování. Do této fáze vstupuje již jen prostý text bez obrázků či formátování. Zde dochází k dalšímu zjednodušení a zkrácení textu aplikací některých běžných postupů:

- Odstranění diakritiky
- Převod veškerého textu na malá písmena
- Převod znaků nového řádku a tabulátoru na mezeru
- Odstranění číslování stránek a pomlček
- Odstranění slov kratších než 3 znaky
- Odstranění stop slov

Po tomto zpracování je výsledkem už pouze jeden dlouhý řádek textu, který je vhodným vstupem pro hashovací funkce.

Důležitým prvkem této fáze je odstranění tzv. *stop slov*. Podle Procházky (2012) se jedná o taková slova, která samy o sobě nenesou žádnou nebo téměř žádnou informaci, která by byla důležitá pro daný účel. Vyskytují se ale velmi často, takže je vhodné mít seznam slov, která lze z textu odstranit a tím vylepšit výkon systému.

Mezi typická česká stop slova patří například následující:

- **Předložky:** na, pro, v, u, k…
- **Spojky:** ani, ale, aby…
- **Zájmena:** jeho, můj, oni, já…
- **Některá slovesa:** mít, být, dělat…

- **Slova, která se často vyskytují v daném kontextu (závislé na konkrétní aplikaci):** diplomová, bakalářská, práce, fakulta, univerzita, tabulka, obrázek, WWW…

Stop slova reflektují zvyky daného přirozeného jazyka – například v Angličtině mezi typická stop slova patří například členy nebo modální slovesa ve všech tvarech.

3. Tvorba hashů
Ve třetí fázi systém vezme předpřipravený text a prochází jej od začátku do konce. Postupně zpracovává všechny pětice slov tak, že je seřadí podle abecedy a vytvoří z nich MD5 hash. Následně se z tohoto hashe vezme 8 bytů a ty se uloží do databáze. Takto postupně vzniknou hashe pro všechny uspořádané pětice slov vstupního textu.

4. Porovnání
Poslední fází zpracování je samotné porovnání dokumentů. To probíhá tak, že se každý uložený hash nově nahraného dokumentu porovná s každým hashem, který byl v databázi uložen dříve.

Pokud jsou nalezeny shodné hashe, znamená to že v dokumentech existují stejné sekvence slov. V takovém případě vzniknou záznamy o podobnosti mezi danými dvěma dokumenty – ten je vyjádřen jako procento shodných hashů. Do databáze jsou pak uloženy dva záznamy – procento podobnosti prvního dokumentu s druhým a procento podobnosti druhého dokumentu s prvním – tyto procenta se mohou vzájemně lišit.

2.9 Další dostupné antiplagiátorské systémy

Ze strany univerzit existuje vysoká poptávka po antiplagiátorských systémech a zdaleka ne všechny z nich jsou schopny realizovat své vlastní řešení. I z tohoto důvodu v posledních několika letech vznikla řada akademických projektů zejména na fakultách se zaměřením na informační technologie, které se zabývají návrhem a implementací antiplagiátorských systémů a jejichž výsledky jsou dále nabízeny k využití dalším univerzitám. Samozřejmě existují i ryze soukromé firmy, které se zabývají vývojem antiplagiátorských systémů pro komerční využití.

Na tomto místě bych rád zmínil některé z těchto systémů:

Turnitin
Jedná se o jeden z nejznámějších antiplagiátorských systémů, které jsou k dispozici. Autoři na webu http://turnitin.com uvádí, že systém Turnitin má pro potřeby vyhledávání plagiátů k dispozici přes 45 miliard webových stránek, 337 milionů prací studentů a 130 milionů článků z akademických knih a publikací. Práce tak lze porovnávat se značným množstvím možných zdrojů.

Za zmínku také stojí, že systém existuje i ve verzi pro tablety a je schopen využívat některá populární online úložiště, jako například Google Drive či Dropbox.

Systém Turnitin však není zaměřen pouze na oblast detekce plagiátů – jedná se o komplexní online službu pro vzdělávací instituce, která nabízí vyučujícím možnost efektivně poskytovat hodnocení a zpětnou vazbu na odevzdané práce či vkládat hlasové komentáře. Rovněž umožňuje studentům vzájemně hodnotit své práce.

Systém obdržel ocenění Editors' Choice webu PC Magazine PCMag.com, který vyzdvihuje intuitivnost a přehlednost a mnoho užitečných funkcí pro vyučující i studenty. V článku je uvedena i cena, která činí 3 dolary na studenta za rok pro vzdělávací instituce.

Ephorus

Ephorus evropský antiplagiátorský systém a autoři na svém webu ephorus.com uvádí, že jej využívá přes 5 000 škol a univerzit. Oproti systému Turnitin je Ephorus zaměřen pouze na detekci plagiátorství a nenabízí další funkce pro vyučující a studenty – alespoň zatím.

Vzhledem k tomu, že Ephorus i Turnitin sdílí společný cíl (zvýšení kvality vzdělání), se tyto systémy spojily a nyní využívají společnou databázi, což dále zvyšuje efektivitu obou systémů.

Cena systému není veřejně uvedena, závisí ale na typu vzdělávací instituce, počtu studentů a délce kontraktu. Stejně tak autoři veřejně neuvádí, na jakém principu přesně systém funguje.

Urkund

Systém Urkund vznikl v roce 2000 ve Švédsku a dnes jej využívají zákazníci v Evropě, Americe i Austrálii. Jedná se čistě o antiplagiátorský systém, který práce porovnává s internetem, vědeckými články v partnerských databázích a texty od samotných studentů. Rovněž je možné do porovnávání zahrnout další databáze, které poskytne samotný klient.

Služby je možné využívat pomocí webu, e-mailu nebo například i systému Microsoft Sharepoint.

Za zmínku stojí, že sami autoři na webu urkund.com uvádí, že systém byl autory v testu antiplagiátorského softwaru (Weber-Wulff, Möller, Touras, Zincke, 2013) vyhlášen jako nejefektivnější software z hlediska nalézání plagiátů.

Přesnou cenu ani podrobnosti o použitých algoritmech autoři veřejně neuvádějí.

Theses

Systém Theses je českým projektem, jehož koordinátorem je Masarykova univerzita. Jedná se o nástroj proti plagiátorům závěrečných prací a využívají jej desítky soukromých i veřejných škol v České Republice i na Slovensku. Systém

jako první nasadila Masarykova univerzita v roce 2006 a další školy se do projektu zapojují postupně.

Autoři na webu theses.cz uvádí, že systém vyhledává podobnosti v databázi dokumentů, která zahrnuje závěrečné práce zapojených škol vědecké publikace v systému Repozitar.cz i další zdroje na internetu.

Rozvoj systému byl v minulosti opakovaně finančně podpořen ze strany Ministerstva školství, mládeže a tělovýchovy České republiky, což svědčí o podpoře tohoto projektu ze strany vlády.

Odevzdej.cz

Systém Odevzdej.cz navazuje na projekt Theses.cz a nabízí službu pro odhalování plagiátů v pracích.

Hlavní rozdíl oproti všem předchozím uvedeným systémům spočívá v tom, že systém Odevzdej.cz může využívat kdokoliv z řad veřejnosti i bez registrace, nikoliv pouze zaměstnanci zapojených institucí.

Podporovanými formáty jsou běžně využívané typy dokumentů systémů Microsoft Office, Open Office i PDF.

EVE2 Plagiarism Detection System

Systém EVE2 je nástroj pro odhalování plagiátů mezi odevzdanými pracemi. Umí pracovat s dokumenty v prostém textu i formáty Microsoft Word a Corel WordPerfect. Zdroje plagiátorství jsou vyhledávány na internetu, oficiální web canexus.com nezmiňuje žádnou vlastní databázi prací.

Pořizovací cena systému je 29,99 dolarů za licenci s neomezeným využitím.

WCopyfind

Systém WCopyfind k slouží porovnání dokumentů a hledání jejich vzájemné shody. Nejedná se o standardní antiplagiátorský nástroj vhodný pro využití na univerzitách – na druhé straně je však tento systém zcela zdarma a dokonce jsou na webu projektu http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/ k dispozici i kompletní zdrojové kódy, čímž případným zájemcům umožňuje detailní analýzu funkčnosti.

StrikePlagiarism

Projekt StrikePlagiarism.com existuje od roku 2002, kdy byl v Polsku spuštěn jako antiplagiátorský nástroj pro využití na vysokých školách. Původně byl poskytován zdarma, dnes je systém placený se slevou pro univerzity. Autoři na webu strikeplagiarism.com bohužel nemají uvedenou přesnou cenu ani technické detaily systému.

Copyscape

Že se plagiátorství zdaleka netýká jen školních či vědeckých prací dokazuje například projekt Copyscape. Služby tohoto systému slouží k odhalování plagiátů mezi webovými stránkami, k čemuž jsou využívány vyhledávače Yahoo! a Google.

Z výčtu, který není ani zdaleka kompletní, je evidentní, že realizace antiplagiátorských systémů je v současnosti velmi intenzivně řešené téma jak na univerzitách, tak i mezi komerčními firmami. Na výběr je mnoho systémů, které se liší cenou i nabízenými možnostmi – od těch, které nabízí pouze základní porovnávání dokumentů mezi sebou až po komplexní e-learningové systémy, které ve kterých je detekce plagiátorství pouze jednou z mnoha funkcí.

3 Vlastní práce

Cílem práce bylo na základě teoretických znalostí a analýzy algoritmu antiplagiátorského systému nalézt jeho nejvhodnější parametry. Dále pak odhalit obecné slabiny používaného řešení a navrhnout vylepšení.

Tato kapitola popisuje proces tvorby dat nutných pro následné testování a statistické vyhodnocení vlivu nastavení systému na rychlost zpracování a přesnost.

3.1 Metodika práce a tvorba dat

Pro přesnou analýzu výkonu algoritmu bylo potřeba vytvořit databázi dokumentů, které budou vhodným vzorkem textu, a zároveň bude předem známa míra podrobnosti mezi jednotlivými dokumenty.

Jako základní nástroj pro tvorbu textů jsem využil generátor textů Lorem ipsum, který je dostupný na http://loripsum.net/ a jehož API umožnilo velmi rychle vygenerovat velké množství vzájemně různých textů, které však zároveň nesou znaky přirozeného textu v přirozeném jazyce.

Dále jsem využil generátor textů Blabot dostupný na http://cs.blabot.net/ – poskytuje poskytující podobné možnosti jako lorem ipsum, který ale navíc generuje texty v Českém jazyce.

Posledním typem textu, který jsem vytvořil, byly dokumenty skládající se pouze z opakujících se několika málo řetězců znaků, u kterých byla jistota, že nebude docházet k náhodným shodám sekvencí slov jako v případě náhodně generovaných dokumentů.

Celkem jsem vytvořil 100 dokumentů, každý o cca 90 tisících znacích, což odpovídá 50 normostranám textu (1 normostrana = 1800 znaků). Polovinu dokumentů jsem uložil jako Portable Document Format (PDF soubor), polovinu jako prostý text (TXT soubor).

Dokumenty byly dále rozděleny na základě vzájemné podobnosti:

- 0%
- 25%
- 50%
- 75%
- 100%

Míra podobnosti udává, kolik mají dva dokumenty shodných vět (respektive slov). Pokud například existují dokumenty A a B nemající žádné společné věty, lze z nich vytvořit dokument C tak, že bude obsahovat 25% vět z dokumentu A a 75% vět z dokumentu B. Pak je evidentní, že podobnost dokumentů A a C je 25% a podobnost dokumentů B a C je 75%.

Za zmínku stojí, že systém Anton uvádí 2 míry podobnosti: Podobnost dokumentu A a dokumentu B a rovněž také podobnost dokumentu B a dokumentu A. Tyto dvě čísla nemusí byt totožná – pokud například existuje dokument A obsahující 1000 slov a dokument B obsahující 2000 slov (prvních 1000 slov dokumentu B jsou slova z dokumentu A, zbytek jsou jiná slova), bude podobnost dokumentu A s dokumentem B 100% (slova dokumentu A je podmnožinou dokumentu B, všechny chunky (chunk je uspořádaná n-tice slov) z dokumentu A se tedy vyskytují v dokumentu B), ovšem podobnost dokumentu B s dokumentem A bude pouze 50% (pouze polovina slov z dokumentu B se vyskytuje v dokumentu A).

Z hlediska určení správné délky chunku je důležitá otázka, jak spojovat dva dokumenty do jednoho, abych získal požadovanou shodu. Nabízí se zde několik možností:

- Spojovat dokumenty slovo po slovu – jedno slovo z prvního dokumentu, druhé z druhého dokumentu
- Vzít první část prvního dokumentu a k ní připojit část druhého dokumentu, bez jakéhokoliv střídání slov
- Spojovat dokumenty po celých skupinách slov

Rozhodl jsem se pro třetí možnost, protože se domnívám, že nejlépe reprezentuje případné plagiátorství. Bral jsem postupně skupiny 30 – 40 slov z každého dokumentu – náhodně, pomocí funkce rand() a vytvářel z nich cílový dokument s požadovanou celkovou shodou.

Výsledky testů (procenta shodnosti) byly extrahovány z databáze a ukládány do CSV souborů spolu s identifikátory dokumentů. Tato procenta byla následně porovnána s cílovou přesností (která byla předem známá, vypočítaná na základě počtu shodných slov) a absolutní hodnota rozdílu se používala dále. Za klíčový údaj o přesnosti jsem považoval průměrnou hodnotu odchylky pro danou délku chunku / hashe.

Porovnávány byly vzájemně pouze texty, které byly vygenerovány stejným zdrojem. Důvod je ten, že dokumenty v různých jazycích nevygenerují nikdy stejné n-tice slov, ani při velmi malých délkách chunku.

3.2 Analýza časové náročnosti algoritmu

Časová náročnost algoritmu je jedna z klíčových vlastností, které je třeba zvážit v souvislosti s využíváním tohoto algoritmu, respektive jeho optimalizací.

Veškeré testování probíhalo na virtuálním PC (s využitím software Oracle VM VirtualBox) s následující konfigurací:
Procesor: 3,20GHz (hostitelské PC Intel Core i5 4570, 4 jádra)
Operační paměť: 4GB (hostitelské PC 8GB)

Operační systém: Kubuntu GNU coreutils 8.5 (hostitelské PC Windows 7 64b)

Jazyk PHP bohužel nativně nepodporuje možnost výpisu toho, kolik procesorového času reálně zabral běh daného skriptu. Bylo tedy třeba čas měřit pomocí podporovaných funkcí. Zvolil jsem standardní PHP funkce DateTime() a MicroTime(), které vrací systémový čas. Funkce jsem zavolal na začátku a na konci skriptu a výsledky od sebe odečetl – tím jsem získal informaci o tom, jak dlouho skript běžel. Jako pojistku jsem použil funkci sys_getloadavg(), abych zjistil průměrnou zátěž systému. Tato funkce vrací průměrný počet procesů, které čekají ve frontě na spuštění.

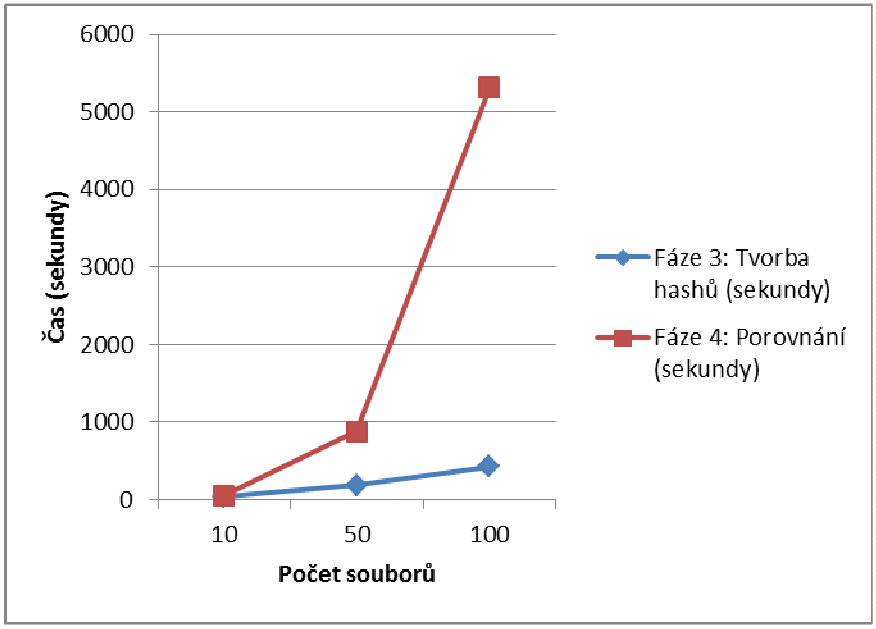
Veškeré měření času probíhalo opakovaně (třikrát), abych eliminoval případný vliv náhodě běžících jiných procesů, které by mohly zatěžovat procesor. Podle sys_getloadavg() však byl počet procesů ve frontě velmi nízký a po celou dobu testování téměř konstantní (1,05 – 1,11), tudíž se domnívám, že lze naměřené hodnoty považovat za dostatečně reprezentativní a přesné. Níže uvedené hodnoty jsou průměrem naměřených hodnot.

3.2.1 Závislost časové náročnosti na počtu zpracovávaných dokumentů

Nejprve jsem analyzoval vliv počtu zpracovávaných souborů na dobu zpracování fází 3 (tvorba hashů) a 4 (porovnání). Výsledky tohoto měření jsou v tabulce a grafu níže – časové údaje jsou uvedeny v sekundách:

Tab. 1 Závislost časové náročnosti na počtu souborů

Počet souborů	**10**	**50**	**100**
Fáze 3: Tvorba hashů (sekundy)	38,19	188,07	427,6
Fáze 4: Porovnání (sekundy)	53,61	876,25	5312,05



Obr. 2 Závislost časové náročnosti na počtu souborů

Z výsledků je patrné, že se zvyšujícím se počtem souborů se čas potřebný pro fázi porovnávání zvyšuje výrazně rychleji, než čas potřebný pro fázi vytváření hashů. Důvod je ten, že zatímco doba potřebná pro vytvoření hashů závisí pouze na počtu slov v daném souboru, při porovnávání je třeba porovnat každý nově vytvořený hash s každým již existujícím hashem v tabulce – kterých je pro každý dokument o 50 normostranách přibližně 11 tisíc.

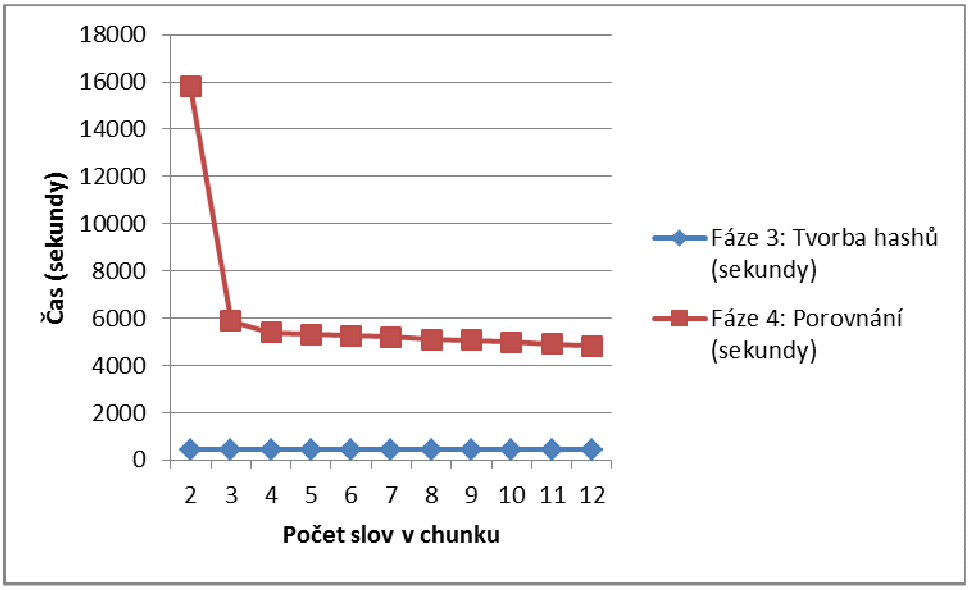
Fáze tvorby hashů je lineárně závislá na počtu zpracovávaných souborů, zatímco fáze porovnávání závislá kvadraticky – trvá tedy výrazně déle, při větším množství souborů je doba trvání fáze 3 v porovnání s fází 4 téměř zanedbatelná. Provádí se pro každý dokument pouze jednou, zatímco fáze porovnání se provádí n-krát (kde n je počet dokumentů v databázi.

3.2.2 Závislost časové náročnosti na počtu slov v jednom chunku

Počet slov v jednom chunku udává, z kolika slov se bude počítat hash. Počet slov v chunku však nijak výrazně nemění počet chunků, které budou během zpracování dokumentu vytvořeny a jejichž hash se bude počítat.

Tab. 2 Závislost časové náročnosti na délce chunku

Délka chunku	2	3	4	5	6	7	8	9	10	11
Fáze 3: Tvorba hashů (sekundy)	427,18	424,5	427,53	427,6	428,11	428,14	428,23	429,33	431,02	433,87
Fáze 4: Porovnání (sekundy)	15813	5871,4	5388,5	5312,1	5255,3	5198,2	5102,4	5064,6	4987,2	4912,4



Obr. 3 Závislost časové náročnosti na délce chunku

Z měření vyplývá, že se zvyšující se délkou chunku se zvyšuje čas potřebný pro výpočet hashe a snižuje se čas potřebný pro porovnání dokumentů.

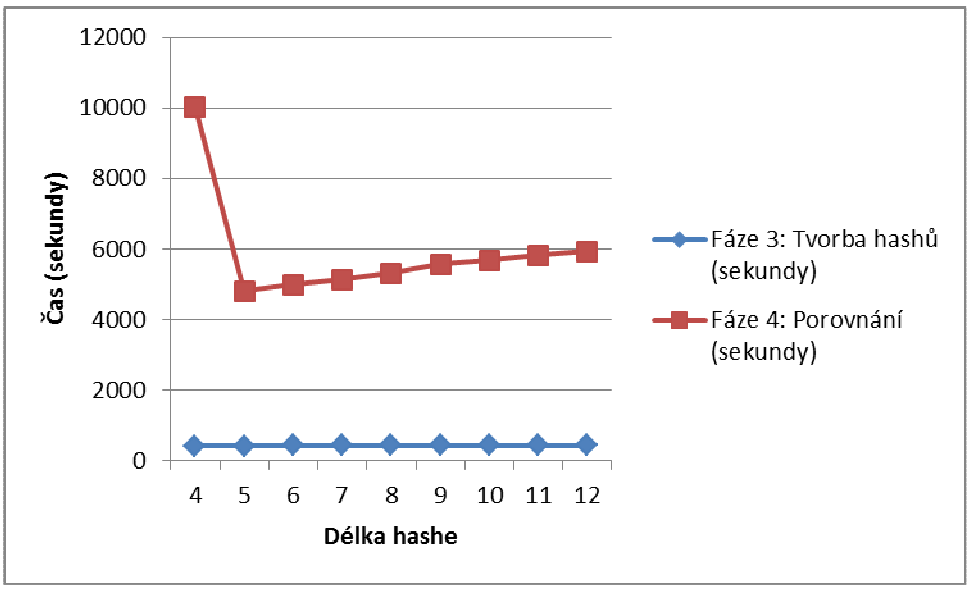
Za zmínku stojí, že při velmi nízkém počtu slov v chunku se výrazně zvýšil čas potřebný pro fázi porovnání – to je způsobeno tím, že kvůli většímu množství shodných hashů skript provádí výrazně více databázových operací.

3.2.3 Závislost časové náročnosti na počtu bytů hashe ukládaných do databáze

Na tomto místě sleduji závislost časové náročnosti na délce hashe (=počtu Bytů MD5 hashe, které se ukládají do databáze a které se následně porovnávají). Výsledky jsou zachyceny v následující tabulce a grafu.

Tab. 3 Závislost časové náročnosti na délce hashe

Délka hashe	4	5	6	7	8	9	10	11	12
Fáze 3: Tvorba hashů (sekundy)	412,4	420,8	424,2	426,3	427,6	429,3	430,1	431,8	433,5
Fáze 4: Porovnání (sekundy)	10018	4812,7	4982,1	5127,3	5312,1	5563,2	5682,7	5815,5	5937,9



Obr. 4 Závislost časové náročnosti na délce hashe

Je evidentní, že se zvyšující se délkou hashe poměrně výrazně roste výpočetní náročnost, zejména ve fázi porovnávání. Porovnání hashů o délce 12 Bytů trvá oproti porovnání hashů o délce 8 Bytů o 12% déle.

Výjimkou jsou opět velmi nízké hodnoty hashe, při kterých se zvyšuje čas potřebný pro fázi porovnání – podobně jako v případě krátkého chunku systém nachází značné množství shodných hashů a kvůli tomu se provádí výrazně více databázových operací.

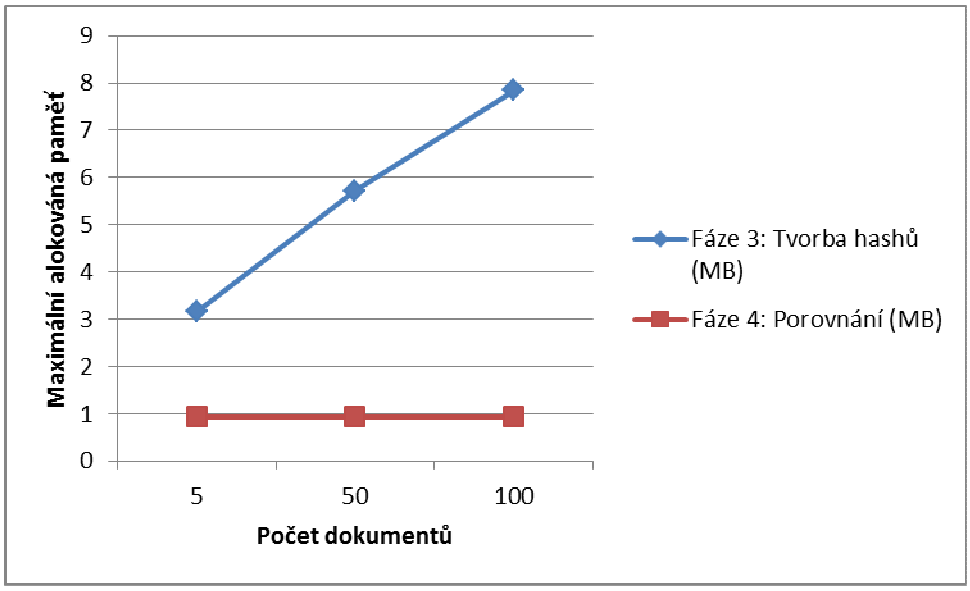
3.3 Analýza paměťové náročnosti algoritmu

Vedle časové náročnosti je náročnost na operační paměť další z významných vlastností algoritmů.

Z tabulky a grafu níže je patrné, že počet souborů má vliv na alkovanou paměť, vývoj je lineární:

Tab. 4 Závislost paměťové náročnosti na počtu souborů

Počet souborů	5	50	100
Fáze 3: Tvorba hashů (MB)	3,153	5,718	7,845
Fáze 4: Porovnání (MB)	0,937	0,937	0,937



Obr. 5 Závislost paměťové náročnosti na počtu souborů

Délka chunku se však ukázala být irelevantní, se změnou počtu slov v chunku se alokovaná paměť nemění, jak je vidět na tabulce níže:

Tab. 1 Závislost paměťové náročnosti na délce chunku

Délka chunku	2	5	6	10	12
Fáze 3: Tvorba hashů (MB)	7,845	7,845	7,845	7,845	7,845
Fáze 4: Porovnání (MB)	0,937	0,937	0,937	0,937	0,937

Stejně tak délka hashe nemá na paměťovou alokaci vliv, jak ukazují čísla v tabulce níže:

Tab. 5 Závislost paměťové náročnosti na délce hashe

Délka hashe	4	8	16
Fáze 3: Tvorba hashů (MB)	7,845	7,845	7,845
Fáze 4: Porovnání (MB)	0,937	0,937	0,937

Paměťová náročnost (maximální alokovaná paměť skriptu, zjištěná pomocí příkazu memory_get_peak_usage() za běhu skriptu) tak zůstává neovlivněna jakýmkoliv nastavením parametrů. Zvyšuje se však s počtem dokumentů. S přihlédnutím k množství paměti, kterým disponují dnešní běžné počítače, včetně těch nejlevnějších v domácnostech, je však paměťová náročnost velmi nízká i pro větší množství zpracovávaných souborů.

3.4 Analýza přesnosti

Přesnost je klíčovou vlastností systému na odhalování plagiátů a v této podkapitole popíši vliv jednotlivých parametrů na její vliv.

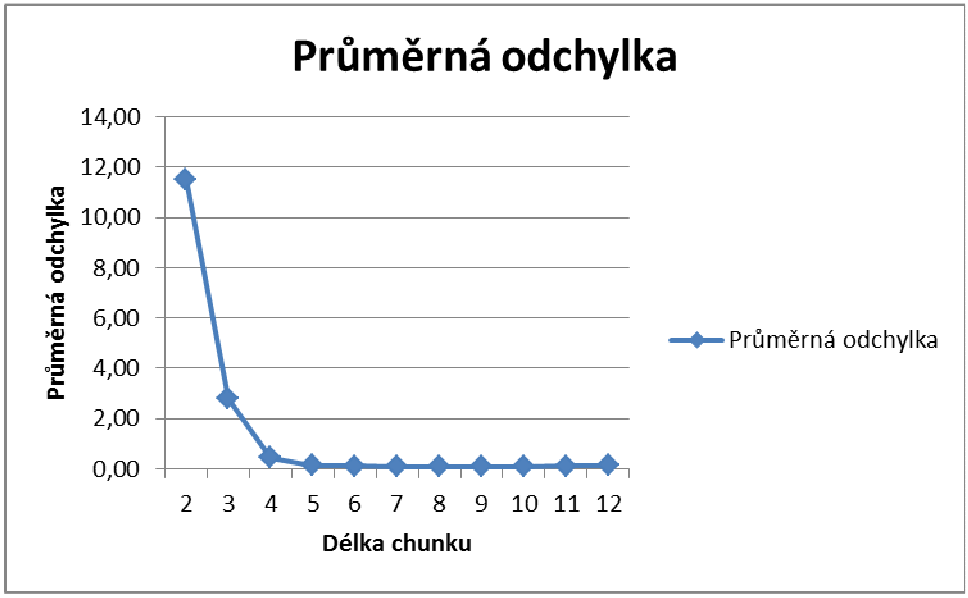
Přesnost jsem měřil jako aritmetický průměr absolutních hodnot rozdílů cílové podobnosti dokumentů (která byla dopředu známá) a míry podobnosti, kterou vypočítal systém Anton. Menší čísla znamenají menší rozdíl cílové a vypočtené podobnosti a tedy větší přesnost.

3.4.1 Závislost přesnosti na délce chunku

Jak moc ovlivní počet slov v jednom chunku přesnost systému ukazuje tabulka a graf níže:

Tab. 6 Závislost odchylky na délce chunku

Chunk	2	3	4	5	6	7	8	9	10	11	12
Podobnost 0%	28,32	5,39	1,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Podobnost 25%	11,03	3,63	0,73	0,36	0,31	0,30	0,23	0,23	0,24	0,27	0,32
Podobnost 50%	10,14	2,52	0,49	0,25	0,18	0,17	0,15	0,15	0,16	0,20	0,26
Podobnost 75%	8,16	2,38	0,03	0,03	0,03	0,03	0,02	0,02	0,03	0,08	0,11
Podobnost 100%	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Průměrná odchylka	11,53	2,78	0,45	0,13	0,10	0,10	0,08	0,08	0,09	0,11	0,14



Obr. 6 Závislost odchylky na délce chunku

Z měření vyplývá, že pro délky chunku 5 – 11 poskytuje systém velmi podobnou přesnost. Dále se zvyšující se počet slov v chunku na přesnosti podepisuje spíše negativně.

Pro hodnoty chunku nižší než 4 se přesnost výrazně snižuje, protože systém nalézá velké množství falešných shod.

Lze očekávat, že v případě plagiátorství se budou nejčastěji opisovat celé věty, nebo dokonce celé odstavce či kapitoly textu. Podle Klimeše (1982) je průměrná délka věty v odborném stylu 8,52 slova a průměrný počet vět v souvětí jsou 2 věty. Pokud tyto údaje odpovídají realitě, lze se domnívat, že pokud počet slov v chunku nepřekročí přibližně 10 – 15, nebude způsobena žádná významná nepřesnost. Zároveň je však třeba mít na paměti, že ne všechna slova z původní věty budou reálně zpracována – pokud systém vhodně odstraní stop slova, počet zpracovávaných slov se zredukuje.

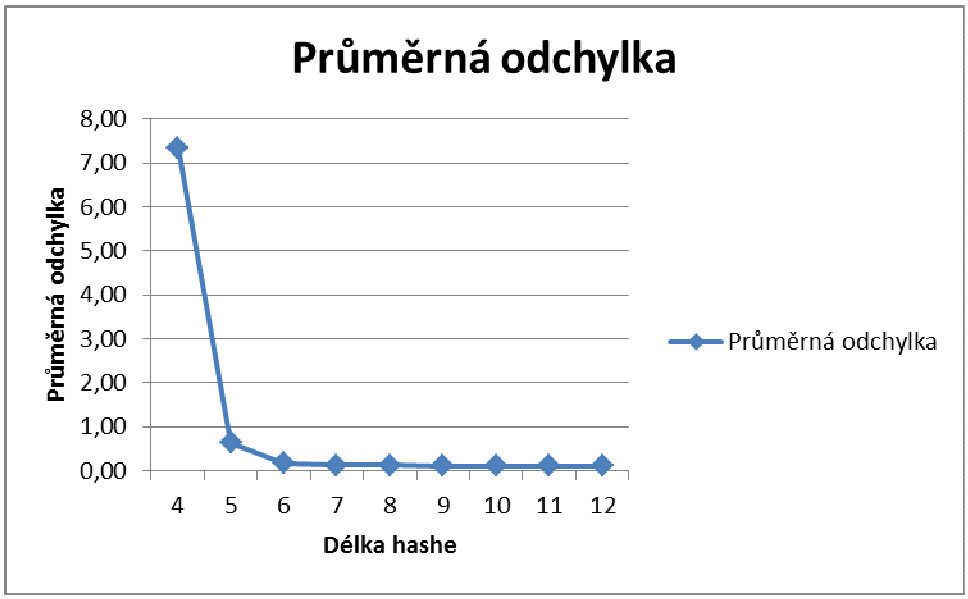
I z tohoto důvodu bych nedoporučoval výrazné zvyšování délky chunku – falešné shody se velmi často objevují pouze u hodnot do 3 – 4, tudíž ve výrazném zvyšování chunku nespatřuji zádný benefit.

3.4.2 Závislost přesnosti na počtu bytů hashe ukládaných do databáze

Jak již bylo zmíněno výše, systém Anton ve třetí fázi spočítá MD5 hash pro každý chunk a část tohoto hashe se uloží do databáze. Závislost přesnosti na počtu bytů MD5 hashe ukazuje tabulka a graf níže:

Tab. 7 Závislost odchylky na délce hashe

Délka Hashe	4	5	6	7	8	9	10	11	12
Podobnost 0%	16,20	1,52	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Podobnost 25%	11,31	1,03	0,53	0,36	0,36	0,36	0,36	0,36	0,36
Podobnost 50%	6,79	0,53	0,28	0,25	0,25	0,25	0,25	0,25	0,25
Podobnost 75%	2,44	0,10	0,02	0,03	0,03	0,03	0,03	0,03	0,03
Podobnost 100%	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Průměrná odchylka	7,35	0,64	0,17	0,13	0,13	0,13	0,13	0,13	0,13



Obr. 7 Závislost časové náročnosti na délce hashe

Z experimentu je zřejmé, že určení správné délky hashe je pro přesnost klíčové. Pro příliš malou délku hashe přesnost výrazně klesá (4 Byty a méně), nicméně zvyšování hashe za určitou hranici nemá na zvýšení přesnosti vliv.

Tento jev lze vysvětlit tím, jak funguje MD5 hash a kombinatorikou. Nejprve je třeba mít na paměti, že hashovací funkce MD5 není prostou funkcí, tudíž neplatí, že pro každé dvě vstupní hodnoty x_1 a x_2 takové, že $x_1 \neq x_2$ existují výstupní hodnoty $f(x_1)$ a $f(x_2)$ takové, že $f(x_1) \neq f(x_2)$. Hashovací funkce přiřazuje neomezené množině vstupních hodnot pouze omezené množství hodnot výstupních a tudíž nutně musí existovat kolize (Dirichletův princip).

Pokud má MD5 hash délku 128 bitů (16 Bytů) a každý tento bit může nabývat hodnot 0 nebo 1 (nebo přepočteno na bity hexadecimálních číslic 00 - FF), pak existuje 2^{128} možných kombinací. Takový počet je zřejmě pro antiplagiátorský systém zbytečný – ovšem zůstává otázkou, jak správně určit minimální možný počet bytů hashe, který by se měl použít.

Jak bylo zmíněno výše, pro jeden dokument o 50 normostranách textu vznikne při zpracování přibližně 11 000 hashů. Je tedy třeba minimalizovat pravděpodobnost výskytu falešných shod. Hash o délce 8 bytů má více než 4 miliardy kombinací – hash o délce 7 bytů má pouze 268 milionů kombinací.

3.4.3 Kombinace délky chunku a délky hashe a jejich vliv na přesnost

Níže uvedená tabulka zobrazuje kombinace vlivu délky hashe a délky chunku na celkovou přesnost srovnávání:

Tab. 8 Kombinace délky chunku a délky hashe a jejich vliv na přesnost

		Délka hashe					
		4	5	6	7	8	9
Délka chunku	2	34,78	13,62	12,38	11,53	11,53	11,53
	3	17,45	3,27	2,97	2,78	2,78	2,78
	4	12,38	1,54	0,48	0,45	0,45	0,45
	5	7,35	0,64	0,17	0,13	0,13	0,13
	6	6,54	0,48	0,11	0,10	0,10	0,10
	7	6,32	0,43	0,11	0,10	0,10	0,10
	8	5,37	0,32	0,10	0,09	0,08	0,08
	9	5,29	0,21	0,10	0,09	0,08	0,08
	10	5,28	0,21	0,10	0,09	0,09	0,09
	11	6,17	0,28	0,12	0,11	0,11	0,11
	12	7,28	0,29	0,16	0,14	0,14	0,14

Z uvedených hodnot vyplývá, že z hlediska přesnosti je optimální kombinací délka chunku 8 – 9 slov a délka hashe 8 – 9 Bytů.

3.5 Závěry měření a diskuze výsledků

Ukázalo se, že časová náročnost algoritmu se významně zvyšuje se zvyšující se délkou hashe a zejména s množstvím dokumentů, které je třeba zpracovat. Naopak počet slov v chunku nemá na čas zpracování významný vliv – počet samotných chunků se totiž mění jen minimálně.

Paměťová náročnost se zvyšuje s množstvím zpracovávaných dokumentů, ovšem zůstává neovlivněna délkou chunku či délkou hashe.

Co se týče vlivu velikosti chunku na přesnost, klíčovým faktorem zůstává to, jak dlouhé opsané pasáže se v textech vyskytují. Pokud budeme předpokládat, že průměrná délka takovéto pasáže je přibližně 16 slov, lze očekávat délka chunku 5 – 16 slov bude vykazovat přibližně stejnou přesnost. Na druhé straně je však třeba si uvědomit, že zvyšování délky chunku nemá významný vliv na časovou náročnost a proto z hlediska časové náročnosti není důvod chunk zvětšovat. Za ideální délku chunku tak na základě mého měření lze označit 8 – 9 slov, což je hodnota, u které lze předpokládat jen velmi málo falešných shod.

Větší vliv na přesnost i na rychlost zpracování má délka hashe. Pro velmi malou délku hashe lze očekávat značné množství falešných shod, což je nežádoucí. S rostoucí délkou hashe se přesnost zvyšuje, ovšem od určité hranice (přibližně 8 – 9 Bytů) nebylo zjištěno významné zpřesnění, které by opodstatnilo zvýšené nároky na dobu zpracování.

Důležitým faktorem pro rychlost zpracování zůstává počet dokumentů, které se mají vzájemně porovnávat. Při vyšším počtu dokumentů významně roste doba zpracování.

Jak bylo zmíněno výše, optimální přesnosti systém dosahuje při kombinaci délky chunku 8 – 9 slov a délka hashe 8 – 9 Bytů.

Podobně jako v oblasti kryptografie či komprese dat tak zůstávají protichůdné cíle v podobě maximální přesnosti a minimálních nároků na výpočetní výkon, přičemž nelze mít oboje a je třeba hledat kompromis. S tím jak dlouhodobě roste výkon a klesá cena hardwaru však zastávám názor, že přesnost by v tomto případě měla být důležitější než výkon. Snížená přesnost bude mít za následek zvýšené nároky na následnou lidskou práci, která bývá dražší, než výpočetní čas.

Na druhé straně zůstává faktem, že procento zlepšení přesnosti se například při zvýšení délky hashe z 8 na 9 Bytů nezvýší o tolik, o kolik se zvýší časová náročnost.

3.6 Možnosti oklamání antiplagiátorských systémů

Při práci se zdroji, například v testu antiplagiátorských systémů (Weber-Wulff, Möller, Touras, Zincke, 2013), jsem narazil na zmínky o možném záměrném oklamání elektronických antiplagiátorských systémů. Rozhodl jsem se tento fenomén prověřit a získat více informací.

Antiplagiátorské systémy zpravidla pracují přímo s jednotlivými slovy, respektive podřetězci textu, které činí základ porovnávacích algoritmů. Pokud se slova či jejich písmena změní, software na to zpravidla není schopen adekvátně reagovat. Podřetězce jsou dále hashovány a následně spolu porovnávány – tento přístup je efektivní z hlediska výpočetního výkonu, nicméně jej lze poměrně snadno oklamat.

Krátký, avšak zajímavý článek na toto téma přinesli (Gillam, Marinuzzi a Ioannou, 2010). Rozebírají v něm některé možné postupy, jak antiplagiátorské systémy oklamat:

Záměna synonym

Záměna synonym je jedním z nejjednodušších způsobů, jak cíleně oklamat antiplagiátorské systémy. Český jazyk nabízí nepřeberné množství různých slov stejného nebo podobného významu a jejich vzájemná záměna vede k velmi odlišným podřetězcům v textu.

Tabulka níže uvádí příklady synonym, které lze vzájemně snadno zaměnit – některá ze synonym umožňují i přeformulování více slov.

Tab. 9 Příklady synonym v českém jazyce

Fráze	Synonymum
Mnoho	Spousta, Množství
Přestože	Třebaže, Navzdory
Někdy	Občas
V minulosti	V uplynulých letech
Zkoumaná oblast	Problematika

I když záměna synonym vyžaduje určitou práci a nelze ji provádět zcela automatizovaně, stále se jedná o poměrně snadný a efektivní způsob, jak z pohledu automatického antiplagiátorského systému poměrně výrazně změnit text.

Záměna znaků

Zaměňovat některé znaky za jiné je snadné a může být prováděno zcela automaticky a velmi snadno. V některých fontech si jsou velmi podobná číslice 1 a písmeno malé l či číslice 0 a písmeno O.

Mnohem zajímavější však může být využívání některých znaků zcela jiného písma – například znaků cyrilice, namísto klasické latinky. Na první pohled od sebe znaky není možné rozeznat, nicméně z hlediska počítačového zpracování se jedná o zcela rozdílné znaky:

Tab. 10 Příklady znaků v latince a v cyrilici

Latinka	Cyrilice
a	а
x	х
e	е
c	с
s	ѕ

Podobných znaků existují desítky a je evidentní, že při čtení textu není možné rozpoznat rozdíl. Zaměnit některé (nebo rovnou všechny) výskyty podobných znaků je přitom triviální proces, který umožňuje drtivá většina běžně používaných textových editorů. Kromě cyrilice lze využít například i symboly řecké abecedy. Tyto znaky, které vypadají na první pohled stejně a díky tomu může snadno dojít k jejich záměně, se nazývají *homoglyfy*.

Přidání znaků s bílou barvou písma

Vložení znaků s bílou barvou písma může představovat další ze snadných, avšak efektivních způsobů, jak z pohledu automatizovaných nástrojů velmi výrazně změnit text.

Existují dva základní přístupy:

- Vložení odstavců matoucího bílého textu do prázdných míst v dokumentu

- Záměna mezer za znaky s bílou barvou písma

V obou případech lze navíc i změnit velikost fontu, takže lze relativně snadno dosáhnout požadovaného výsledku a velmi výrazně tak pro detekční nástroje změnit text, i když se na první pohled nezměnil vůbec.

Tab. 11 Příklady znaků v latince a v cyrilici

Původní text	Text, kde byly mezery nahrazeny bílými znaky
Ema mele maso	Ema mele maso
Na pohled stejný text	Na pohled stejný text

Uvedené příklady ukazují, že změnit text tak, aby byl pro člověka totožný a pro software rozdílný, není problém. Nemusí se přitom jednat pouze o záměnu synonym či slovosledu – k dispozici jsou i další, pro plagiátora mnohdy jednodušší a zároveň efektivnější metody.

Uvedeno bylo pouze několik příkladů – je pravděpodobné, že existují desítky dalších metod, jak přidat / změnit text tak, aby si této změny lidský čtenář nevšiml, ale software ji našel a vyhodnotil jako jiný text. Možnosti se také liší mezi jednotlivými formáty – například prostý textový soubor ve formátu TXT mnoho možností nenabízí. Naproti tomu dokumenty ve formátu Microsoft Word mohou obsahovat záhlaví a zápatí, textová pole, různé vrstvy a další způsoby, jak do dokumentu ukrýt text. Stačí pouhá kombinace změny barvy a velikosti písma, a do dokumentu lze ukrýt téměř neomezené množství klamavého textu. Metodami skrývání zpráv se zabývá celá vědní disciplína steganografie – studiem různých jejích metod by jistě bylo možné odhalit mnoho dalších způsobů, jak text ukrýt. K efektivnímu oklamání antiplagiátorských systémů je nutná znalost principu jejich fungování – dle dostupné literatury (viz výše) se však zdá, že většina existujících řešení funguje na principu hashování podřetězců textu. Lze tudíž očekávat, že většina přístupů bude poměrně univerzální.

3.6.1 Testy shody záměrně pozměněných dokumentů

Přirozeně mě zajímalo, jak si s uvedenými způsoby oklamání antiplagiátorského software poradí systém Anton.

Rozhodl jsem se tedy provést několik testů a zjistit, zda systém bude schopen odolat záměrným pokusům o oklamání a správně spočítat procento shodnosti dvou dokumentů.

Text s doplněnými bílým znaky

V tomto testu jsem se rozhodl otestovat, jak si Anton poradí s textem, který obsahuje doplněné bílé znaky. Bílými znaky mám na tomto místě na mysli písmena / slova s bílou barvou fontu (tudíž na bílém pozadí neviditelná) – nikoliv tzv. *white spaces*, což jsou prázdná místa (například znak mezera).

Pro tento test jsem vytvořil 3 dokumenty:

- Dokument A obsahující náhodný text vygenerovaný pomocí lorem ipsum
- Dokument B obsahující tentýž text, ovšem doplněný o stejné množství jiného textu
- Dokument C obsahující tentýž text, ovšem každá druhá mezera je zaměněna za znak malé *l*, které má bílou barvu fontu

Nechal jsem systém porovnat tyto tři dokumenty, výsledky jsou v tabulce níže:

Tab. 12 Analyzované dokumenty a jejich shoda

Dokumenty	Procento shody
B a A	48,5%
C a A	0%

Z provedených testů je evidentní, že pro systém neexistuje rozdíl mezi textem s bílou barvou fontu (tudíž neviditelným) a textem s viditelnou barvou fontu. I bílý text je tak součástí porovnávání, čímž lze systém relativně snadno oklamat. Stačí přidat náhodně vygenerovaný text.

Pokud se textem v bílé barvě nahradí některé mezery v textu, lze dosáhnout dokonce i nulové shody, přestože jsou jinak oba texty identické.

Text se zaměněnými znaky

V tomto testu jsem se rozhodl otestovat to, jak si Anton poradí s textem, ve kterém byla některá písmena záměrně zaměněna za písmena jiná. Jak již bylo zmíněno výše, některá písmena v cyrilici nebo v řecké abecedě vypadají stejně jako písmena v klasické latince, nicméně z hlediska softwarového zpracování jsou odlišná.

Pro tento test jsem vytvořil 2 dokumenty:

- Dokument A obsahující náhodný text vygenerovaný pomocí lorem ipsum
- Dokument B obsahující tentýž text, ovšem písmena *a*, *e*, *c, i* a *s* byla zaměněna za jejich ekvivalenty v cyrilici

Nechal jsem systém porovnat tyto dva dokumenty, výsledky jsou v tabulce níže:

Tab. 13 Analyzované dokumenty a jejich shoda

Dokumenty	Procento shody
B a A	0%

Z provedených testů je evidentní, že systém pracuje s datovou reprezentací. Pokud dva znaky vypadají stejně, nicméně jejich datová reprezentace je odlišná, systém je vyhodnotí jako dva různé znaky. Takto lze během několika málo minut

získat dokument, který antiplagiátorský systém vyhodnotí jako zcela původní, přestože je identický s již existujícím dokumentem.

3.6.2 Závěr testů oklamání antiplagiátorského systému

Testy zaměřené na záměrné oklamání systému Anton potvrdily mé očekávání, že tento software nedokáže správně rozpoznat uměle vytvořenou změnu textu a adekvátně na ni reagovat. Systém se tak čistě z pohledu dat chová správně – různé znaky jsou reprezentovány různě, tudíž nejsou považovány za identické. Je však třeba se zamyslet nad tím, zda je toto chování ideální z hlediska odhalování plagiátů – osobně se domnívám, že nikoliv.

S tímto problémem se však patrně nepotýká zdaleka pouze antiplagiátorský systém Anton – v testu antiplagiátorského softwaru (Weber-Wulff, Möller, Touras, Zincke, 2013) autoři zmiňují, že z celkem 15 testovaných systémů byl pouze Turnitin schopen správně rozpoznat homoglyfy, nahradit je znaky z latinky a identifikovat plagiátorství. Jedná se tedy zřejmě o relativně běžný nedostatek podobných systémů.

Lze se domnívat, že pokusy o záměrné oklamání jsou ze strany autorů tohoto typu software částečně opomíjeny a jedná se tak o relativně neprobádanou oblast.

3.7 Slabiny systému a možná vylepšení

Na tomto místě bych rád zmínil některé slabiny současného řešení, která podle mého názoru stojí za zmínku.

Největší prostor pro vylepšení vidím ve fázi předzpracování dokumentu. Z lingvistického hlediska by například bylo vhodné, kdyby systém byl schopen detekovat a automaticky vhodně zaměňovat synonyma – stal by se tak odolnějším proti drobným úpravám textu. V současnosti se analyzují pouze samotná slova bez jakéhokoliv kontextu a případná možnost záměny synonym se neřeší.

Dalším zajímavým námětem na rozšíření systému by mohla být zvýšená odolnost proti některým pokusům o oklamání systému – například detekce bílého fontu či záměna homoglyfů.

Podobně by bylo možné zvážit morfologickou analýzu slov a pomocí odstranění předpon a přípon slova zaměňovat za jejich základní tvary. Český jazyk má však značné množství lingvistických pravidel a výjimek, takže je otázka, jaká by byla celková efektivita podobných úprav.

Za zmínku taktéž stojí, že současný antiplagiátorský systém ignoruje obrázky a neobsahuje ani žádnou formu analýzy tabulek. Detekce netextových plagiátů je tak zcela opomíjena, což lze však zejména u obrázků vysvětlit pravděpodobně tím, že analýza obrazových dat je obecně náročnější proces než v případě dat textových a je otázkou, nakolik by bylo toto rozšíření výhodné.

Další slabinou všech nejběžnějších antiplagiátorských systémů je otázka porovnávání textů v různých jazycích. Přestože existují články zabývající se

touto problematikou, mnohé z nich mluví zejména o obtížnosti této úlohy. Za všechny články zabývající se touto problematikou bych rád jmenoval například článek pana Baileyho (2011). Existují však i texty nabízející možná řešení – například práce pánů Češky, Tomana a Ježka.

Velmi důležitým rozšířením by však mohla být integrace s dalšími podobnými systémy a jejich databázemi.

Rozšíření antiplagiátorského systému Anton o schopnost vyhledávání pasáží na internetu ve své práci velmi dobře popisuje Veselý (2013) ve své diplomové práci. Využívá internetový vyhledávač Seznam.cz, pomocí kterého hledá fragmenty textu na webu. Vzhledem k tomu, jak na internetu přibývají snadno dostupné materiály (například Google Books nebo veřejně dostupné archivy závěrečných prací fakult), lze toto rozšíření považovat za klíčové.

4 Závěr

Tato práce byla zaměřena na analýzu a optimalizaci antiplagiátorského systému Anton, který je k dispozici na Mendelově univerzitě v Brně.

Úvodní část práce představuje úvod do problematiky, popisuje teoretická východiska a zavádí základní pojmy z oblasti plagiátorství. Rovněž byl zmíněn legislativní rámec a popsán rozdíl mezi plagiátorstvím a jinými formami přejímání cizí práce. Tato část práce také obsahuje popis obecných postupů a technik, které se v antiplagiátorských systémech využívají.

V praktické části byl popsán návrh a metodika testů – konkrétně výpočetní náročnosti, paměťové náročnosti a přesnosti. Dále pak tato kapitola obsahuje shrnutí jejich výsledků a jsou diskutovány závěry, z nich vyplývající doporučení a možná vylepšení. Stejně jako v mnoha dalších oblastech informačních technologií je i v oblasti antiplagiátorských systémů nutné hledat vhodný kompromis mezi výpočetní náročností a kvalitou výstupů. Také byly v této části zmíněny některé slabiny systému a navržena možná vylepšení a rozšíření, kterými by bylo možné se zabývat v budoucnosti. Rovněž byly představeny některé metody, jak antiplagiátorský systém záměrně oklamat.

Jak dokazují mnohé mediálně známé případy plagiátorství z posledních let, je tato problematika velice aktuální. Jednou z nejcennějších věcí, kterou každá univerzita má, je totiž dobré jméno – které může podezření na plagiátorství studentů nenávratně poškodit. Získat dobrou pověst trvá mnoho let a její ztráta může přijít velice rychle. Proto je aktivní přístup k potírání plagiátorství velmi důležitý. Nemělo by však zůstat jen u softwarových řešení na detekci již odevzdaných plagiátů – neméně důležitá je komplexní antiplagiátorská politika zahrnující i poučení studentů a jejich motivaci k poctivé práci.

5 Literatura

BAILEY J. *The Problem with Detecting Translated Plagiarism*. [online]. 2011. [cit. 2014-12-22] Dostupné z WWW: <https://www.plagiarismtoday.com/2011/02/24/the-problem-with-detecting-translated-plagiarism/>

CHANCHAL, K. R., CORDY J. R.: *A Survey on Software Clone Detection Research*. [online]. School of Computing, Queen's University at Kingston, Ontario, Canada, 2007 [cit. 2014-09-12]. Dostupné z WWW: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.7869>

CLOUGH, P.: *Plagiarism in natural and programming languages: an overview of current tools and technologies*. [online]. Department of Computer Science, University of Sheffield, 2000, [cit. 2014-08-27] Dostupné z WWW: <http://ir.shef.ac.uk/cloughie/papers/plagiarism2000.pdf>

Copyscape Plagiarism Checker – Duplicate Content Detection Software [online]. [cit. 2014-12-22]. Dostupné z <http://www.copyscape.com/>

ČERNOHLÁVKOVÁ, K.: *Plagiátorství na vysokých školách* [online]. 2008 [cit. 2014-12-22]. Diplomová práce. Masarykova univerzita v Brně, Filozofická fakulta. Vedoucí práce Petra Šedinová. Dostupné z WWW: <http://is.muni.cz/th/109574/ff_m/diplomova_prace.pdf>

Česká terminologická databáze knihovnictví a informační vědy (TDKIV) [online]. Praha : Národní knihovna České republiky, 2014 [cit. 2014-07-27]. Dostupné z WWW: <http://aleph.nkp.cz/cze/ktd>

Česko. *Autorský zákon (zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů)* [online]. [cit. 2014-07-29]. Dostupné z WWW: <http://business.center.cz/business/pravo/zakony/autorsky/>

Česko. *Trestní zákon (zákon č. 140/1961 Sb.)* [online]. [cit. 2014-07-29]. Dostupné z WWW: <http://business.center.cz/business/pravo/zakony/trestni_zakon/>

ČEŠKA, Z.: *THE FUTURE OF COPY DETECTION TECHNIQUES*. [online]. Západočeská univerzita v Plzni, 2010, [cit. 2014-08-27] Dostupné z WWW: <http://textmining.zcu.cz/publications/YRCASpaper.pdf>

ČEŠKA Z., TOMAN M., JEZEK K. *Multilingual Plagiarism Detection*. [online]. Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia. 2008. [cit. 2014-12-25] Dostupné z WWW: <http://www.researchgate.net/publication/221655858_Multilingual_Plagiarism_Detection

EVE2 Plagiarism Detection for Teachers [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://www.canexus.com/>

EVROPSKÁ KOMISE. *Evropa 2020*. [online]. 2010. [cit. 2014-12-22] Dostupné z WWW: < http://ec.europa.eu/europe2020/index_cs.htm>

FOLTÝNEK T., RYBIČKA J., DEMOLIOU C.: *Do students think what teachers think about plagiarism?* [online]. International Journal for Educational Integrity, 2013, [cit. 2014-12-22] Dostupné z WWW: <http://www.ojs.unisa.edu.au/index.php/IJEI/article/view/931>

GILLIAM L., MARINUZZI J., IOANNOU P. *TurnItOff: defeating plagiarism detection systems*. [online]. 11th Higher Education Academy-ICS Annual, 2010, [cit. 2014-12-22] Dostupné z WWW: <http://www.ojs.unisa.edu.au/index.php/IJEI/article/view/931>

HARRIS, R. Anti-Plagiarism Strategies for Research Papers. 2013 [online]. Dostupné z WWW: < http://www.virtualsalt.com/antiplag.htm>.

Home - Ephorus [online]. [cit. 2014-12-22]. Dostupné z WWW: <https://www.ephorus.com/>

iParadigms Turnitin Review & Rating [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://www.pcmag.com/article2/0,2817,2465541,00.asp>

KLIMEŠ, L.: *Podnětné dílo české kvantitativní lingvistiky*. [online]. ÚJČ Akademie věd ČR, 1982, [cit. 2014-11-30] Dostupné z WWW: <http://nase-rec.ujc.cas.cz/archiv.php?art=6302>

NĚMEČKOVÁ, L.: *Plagiátorství*. [online]. ÚSTŘEDNÍ KNIHOVNA ČVUT, 2009 [cit. 2014-09-12]. Dostupné z WWW: <http://knihovna.cvut.cz//administrace/upload_dir/files/92d3b80c1ab35ca5a6cba67ff8dceaf9c9931380.pdf>

Odevzdej.cz – Odhalování plagiátů v seminárních pracích [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://odevzdej.cz/>

PROCHÁZKA, D.: *SEO: cesta k propagaci vlastního webu*. [online]. Grada Publishing a.s., 2012, [cit. 2014-11-27] Dostupné z WWW: <https://books.google.cz/books?id=KSN0AgAAQBAJ>

ŘEHŮŘEK, R.: *Plagiarism Detection through Vector Space Models Applied to a Digital Library*. [online]. Faculty of Informatics, Masaryk University, 2011, [cit. 2014-11-27] Dostupné z WWW: <http://www.fi.muni.cz/usr/sojka/download/raslan2008/4.pdf>

STEIN, B.: *Fingerprint-based Similarity Search and its Applications*. [online]. Bauhaus-Universität Weimar, 2008, [cit. 2014-11-27] Dostupné z WWW: <http://www.billingpreis.mpg.de/15632/stein.pdf>

Strikeplagiarism.com antiplagiarism system [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://strikeplagiarism.com/>

TETSUYA N., KENSUKE B., YASUHIRO Y., DAISUKE I.: *Partial Plagiarism Detection Using String Matching with Mismatches*. [online]., 2011, [cit. 2014-11-27] Dostupné z WWW: <http://link.springer.com/chapter/10.1007%2F978-3-642-25483-3_21>

The Plagiarism Resource Site - WCopyfind [online]. [cit. 2014-12-22]. Dostupné z WWW: http://plagiarism.bloomfieldmedia.com/z-wordpress/software/wcopyfind/>

Turnitin - Home [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://turnitin.com/en_us/home>

URKUND - Home [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://www.urkund.com>

VESELÝ, O.: *Analýza podobnosti závěrečných prací a online dokumentů* [online]. 2013 [cit. 2014-12-22]. Diplomová práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta. Vedoucí práce Tomáš Foltýnek. Dostupné z WWW: <https://is.mendelu.cz/auth/lide/clovek.pl?id=20645;zalozka=7;studium=41813;zp=30905>

Vysokoškolské kvalifikační práce [online]. [cit. 2014-12-22]. Dostupné z WWW: <https://theses.cz/>

Weber-Wulff D., Möller Ch., Touras J., Zincke E.: *Plagiarism Detection Software Test 2013* [online]. [cit. 2014-12-22]. Dostupné z WWW: <http://plagiat.htw-berlin.de/software-en/test2013/report-2013/>

Zákon o vysokých školách (zákon č. 111/1998 Sb.) [online]. [cit. 2014-07-29]. Dostupné z WWW: <http://www.msmt.cz/vzdelavani/uplne-zneni-zakona-c-111-1998-sb-o-vysokych-skolach-text-se-zapracovanymi-novelami>