

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anej Bandelj

**Odkrivanje plagiatorstva v programski kodi
in prostem tekstu**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge: *Odkrivanje plagiatorstva v programski kodi in prostem tekstu*

Zaradi lahko dostopnih virov na Svetovnem spletu in razvoja komunikacijske tehnologije, se raven plagiatorstva dviguje. Prav tako pa se razvijajo tudi različne računalniške metode za iskanje plagiatov. Danes obstajajo številna orodja za ugotavljanje plagiarizma v prostem tekstu in programski kodi. V okviru diplomske naloge predstavite problem plagiatorstva in analizirajte orodja za odkrivanje plagiatov v prostem tekstu in programski kodi. V zaključku podajte ugotovitve in priporočila za uporabo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani **Anej Bandelj**, z vpisno številko **63070240**, sem avtor diplomskega dela z naslovom:

Odkrivanje plagiatorstva v programski kodi in prostem tekstu

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Aljaža Zrneca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 24. junija 2014

Podpis avtorja:

Zahvala

Zahvaljujem se mentorju viš. pred. dr. Aljažu Zrnecu za pomoč in vodenje pri izdelavi diplomskega dela.

Hvala staršem za podporo pri študiju.

Anej Bandelj

Kazalo

Povzetek	1
Abstract	2
1 Uvod	3
2 Plagiatorstvo	5
2.1 Zgodovina plagiatorstva	5
2.2 Vrste in posledice plagiatorstva	6
2.2.1 Plagiatorstvo na splošno	6
2.2.2 Plagiatorstvo v šolstvu	7
2.3 Kaj izključuje plagiatorstvo?	8
2.3.1 Samoplagiatorstvo	8
2.3.2 Dvojna stvaritev	8
2.3.3 Nezavedno prevzemanje	8
2.3.4 Inspiracija ali parodija	9
2.4 Kazenskopравни vidiki plagiata	9
2.5 Kako omejiti plagiatorstvo?	11
3 Avtorske pravice	12
4 Odkrivanje plagiatorstva v programski kodi	13
4.1 Orodja za detekcijo plagiatov v programski kodi	14
4.1.1 MOSS	15
4.1.2 SIM	18
4.1.3 JPlag	20
4.1.4 Marble	24
4.1.5 Povzetek zgoraj opisanih orodij za detekcijo v programski kodi	27
4.1.6 Priporočilo	29
5 Odkrivanje plagiatorstva v prostem tekstu	30
5.1 Programi in storitve za detekcijo plagiatov v prostem tekstu	30
5.1.1 Plagiarism check.org	31
5.1.2 Plagiarism checker	32

5.1.3	PlagScan.....	33
5.1.4	Viper	37
5.1.5	Plagiarism Detector.....	39
5.1.6	Primerjava rezultatov plagiatorstva med storitvami ter programi	41
5.1.7	Priporočilo.....	43
6	Zaključek	44
	Literatura in viri	46

Kazalo tabel in slik

KAZALO TABEL:

Tabela 1: Primerjava orodij za odkrivanje plagiatorstva v programski kodi. [23]	28
Tabela 2: Primerjava programov in storitev za odkrivanje plagiatorstva v prostem tekstu.....	42

KAZALO SLIK:

Slika 1: Stopnje plagiatorstva v programski kodi [15].	14
Slika 2: Poganjanje sistema MOSS.	16
Slika 3: MOSS prikaz rezultatov.	17
Slika 4: MOSS podrobnejši prikaz podobnosti.....	17
Slika 5: SIM dvostolpični izpis podobnosti programske kode.	19
Slika 6: SIM primerjava datotek med seboj.	19
Slika 7: SIM primerjava datotek z izbrano datoteko.....	20
Slika 8: JPlag poganjanje klienta.....	22
Slika 9: JPlag prikaz rezultatov.....	23
Slika 10: JPlag podrobnejši prikaz rezultatov.	23
Slika 11: Marble normalizacija datotek.	25
Slika 12: Marble pregled o vsebovanosti plagiatorstva.....	25
Slika 13: Marble datotečni izpis o vsebovanosti plagiatorstva.	25
Slika 14: Marble podrobnejši prikaz podobnosti.	26
Slika 15: Plagiarism check.org prikaz poročila o ujemanju besedila na internetu.	32
Slika 16: Plagiarism checker prikaz poročila o ujemanju besedila na internetu.	33
Slika 17: PlagScan prikaz ravni plagiatorstva.....	34
Slika 18: PlagScan podrobnejši prikaz ujemanja našega besedila z besedilom na svetovnem spletu.	35
Slika 19: Prikaz našega dokumenta vključno z obarvanimi stavki, ki se štejejo za plagiatorstvo.	36
Slika 20: Viper poročilo o plagiatorstvu.....	38
Slika 21: Tortni prikaz plagiatorstva, citiranja in našega lastnega besedila.	40
Slika 22: Prikaz, na katerih URL naslovih se pojavlja identično besedilo, kot ga imamo mi v tekstu.	40

Seznam uporabljenih kratic in simbolov

AIPA	Zavod za uveljavljanje pravic avtorjev, izvajalcev in producentov avdiovizualnih del Slovenije
HTML	Hyper Text Markup Language – označevalni jezik za oblikovanje večpredstavnostnih dokumentov
IPF	Zavod za uveljavljanje pravic izvajalcev in proizvajalcev fonogramov Slovenije
KZ	Kazenski zakonik
RUL	repozitorij Univerze v Ljubljani
SAZAS	Združenje skladateljev in avtorjev za zaščito avtorske pravice Slovenije
URL	Uniform Resource Locators – enolični krajevnik vira oz. naslov spletnih strani v svetovnem spletu
ZAMP	Združenje avtorjev Slovenije
ZASP	Zakon o avtorskih in sorodnih pravicah

Povzetek

V zadnjem času pogosto slišimo očitke, da je neko delo plagiat, zato sem se odločil, da v diplomskem delu podrobneje opišem to področje. Najprej bom opredelil, kaj pravzaprav pomeni pojem plagiatorstvo, na katerih področjih je prisotno, kako ga omejiti in kako si s pomočjo računalniških orodij lahko pomagamo pri njegovem odkrivanju. Poglobil sem se v delovanje orodij, ki odkrivajo podobnosti v programski kodi in prostem tekstu. Ta orodja ne določijo samih plagiatov, temveč pridobijo zgolj odstotek podobnosti besedila ali programske kode, ki ga kasneje lahko obravnavamo kot plagiatorstvo. Avtorska dela so individualne intelektualne stvaritve s področja književnosti, znanosti in umetnosti, ki so na kakršen koli način izražene. Avtorske pravice delimo na materialne, moralne in ostale. V primeru plagiatorstva gre torej za kršenje avtorskih pravic. Če se ti očitki izkažejo za utemeljene, takega avtorja v večini primerov doleti moralna obsodba ter s tem povezane sankcije, ki so določene z avtonomnimi pravili znanstvene institucije ali profesionalnega združenja. Še dandanes je zelo težko določiti ali gre za plagiat ali ne, saj običajno razpolagamo z omejenim številom podatkovnih zbirk, veliko pa je tudi takih, ki prevedejo avtorsko delo iz tujega jezika v materni jezik in s tem zmanjšajo možnosti odkritja plagiata. Ravno tako je pri umetnosti zelo težko določiti, ali gre za plagiat ali ne, saj se umetniki zgledujejo drug po drugem. Nekateri se tega dejanja zavedajo, drugi to počnejo povsem nezavedno. Plagiatorstvu se najbolje izognemo tako, da pri dobessednem navajanju ali povzemanju besedila nekega drugega avtorja vedno uporabimo ustrezno citiranje ter navedemo vir, kjer je bilo delo objavljeno.

Ključne besede: plagiatorstvo, intelektualne stvaritve, avtorske pravice, odkrivanje plagiatorstva, citiranje

Abstract

Lately we often hear allegations that a certain work is a plagiarism, hence I decided to describe this area in detail in my diploma thesis. First I define what precisely the term plagiarism means, in which areas it is present, how to limit it, and how to utilize software for its detection. I delve into the utility of software which detects details in source code and text documents. Such software does not determine plagiarism itself, but rather indicates the percentage of text or source code similarity, which we can subsequently treat as plagiarism. Original works are individual intellectual creations from the fields of literature, science, and art expressed in any form. Authors' rights are divided into economic rights, moral rights, and other rights. Plagiarism therefore constitutes a violation of authors' rights. Should such allegations prove to be substantiated, the author in question is in most cases subject to moral judgment and other related sanctions laid down in the autonomous rules of a scientific institution or professional association. It is, however, at present exceedingly difficult to ascertain whether a work is a plagiarism or not, since we normally have a very limited number of databases at our disposal, and often an original work is translated from a foreign language into one's native language, thus reducing the possibilities of detecting plagiarism. It is also very difficult to determine plagiarism in art, since artists imitate one another. Some of them are aware of their actions, while others are entirely unaware. Plagiarism is best avoided by always using the appropriate citation style and stating the source where a work was published when quoting or paraphrasing an author.

Key words: plagiarism, intellectual creations, authors' rights, plagiarism detection, citation

1 Uvod

Plagiatorstvo (tudi plagiarizem) je v slovarju razloženo kot dejanje ali primer uporabe, ko oseba natančno posnema jezik in miselnost drugega avtorja brez njegovega dovoljenja ter zastopa to avtorsko delo kot svojo lastnino [40]. Torej gre za nekorekten odnos do intelektualne lastnine ter njeno prisvajanje, saj je prevzemanje in objavljajanje tujega dela pod svojim lastnim imenom brez ustreznih navedb prepovedano. Do tega lahko pride namerno kot kraja ali nenamerno, v večini primerov zaradi nepoznavanja in iz površnosti.

Beseda plagiat je v slovarju Slovenskega knjižnega jezika obrazložena kot "kar je prepisano, privzeto od drugod in objavljeno, prikazano kot lastno, navadno v književnosti" [36]. Torej se najpogosteje pojavlja kot kraja informacij, ki jih je druga oseba že napisala. Lahko pa se plagiatorstvo pojavi tudi pri dobesednem govoru ali kopiranju ideje druge osebe, ki jo kasneje predstavimo kot svojo [13, 14, 19, 27]. To vključuje informacije iz spletnih strani, knjig, pesmi, televizijskih oddaj, elektronskih sporočil, intervjujev, člankov, umetniških del ali katerih koli drugih medijev. Kadar koli razlagamo z drugimi besedami, povzemamo besedne zveze ali stavke iz dela nekoga drugega, moramo vedno navesti vir informacij v našem dokumentu oz. uporabimo citiranje. Citiranje pomeni dobesedno navedeno oziroma prepisano besedilo iz drugega avtorskega dela [10]. Dobesedno citiramo, kadar je besedilo kopirano, ali so bile uporabljene ideje, ki so delo drugega avtorja. Torej ni dovolj, da dodamo povezavo na vir zgolj na koncu našega članka. V primeru, da ni navedenega vira ali ni označeno citiranje za že obstoječe informacije, se delo šteje za plagiat.

Zaradi hitrega napredka tehnologije, interneta in vse bolj izpopolnjenih spletnih iskalnih algoritmov, imamo na razpolago ogromno količino informacij. Zato se vse manj odločamo za iskanje informacij iz knjig, revij, časopisov, saj je večina informacij že na voljo na internetu v elektronski obliki, čeprav niso vedno relevantne in verodostojne. Internet je postal največje javno skladišče informacij in komunikacij, ki se nenehno dopolnjuje. Ker je informacij zelo veliko, so tudi take, ki se ponavljajo in so vsebovane na več spletnih straneh. Do njih lahko dostopa tako rekoč vsak, saj ni načina, ki bi uporabnikom to preprečeval. Tako pridemo do problema, saj nam internet daje vse možnosti za izvajanje plagiatorstva. Nad kopiranimi viri pa je zelo težko določiti njihov izvor in zanesljivost, saj ni poostrelega nadzora nad njihovim kopiranjem in razmnoževanjem. Plagiatorstvo se po mnenju raziskovalcev dandanes zelo povečuje, kar predstavlja resen problem v izobraževanju ter na ostalih področjih.

Večkrat sem se spraševal, kako so lahko profesorji v času mojega študija pri tako velikem številu študentov pregledovali seminarske in ostale naloge. Rezultati, ki so nam jih podali, so bili v veliki meri zelo zanesljivi, saj so v primeru prepisovanja naloge to izredno hitro odkrili.

To mi je dalo še dodatno spodbudo, da sem se odločil napisati diplomsko delo povezano s to tematiko.

V diplomskem delu predpostavljam, da je pri preverjanju programske kode le-ta napisana v programskem jeziku Java, saj sem tako najlažje primerjal orodja med seboj. Java je objektno usmerjen programski jezik, ki ga je razvil James Gosling v podjetju Sun Microsystems in je za razvijalce na voljo že od leta 1995. Java je pridobila zaupanje številnih programerjev, poleg tega pa še danes predstavlja znanje programiranja v Javi eno najboljših priporočil za zaposlitev.

Pri opisih plagiatstva se bom omejil na plagiatstvo med študenti fakultet naravoslovnih ved. Čeprav je zaradi hitrejšega napredka v naravoslovju in tehniki bistveno manj plagiatov kot v družboslovju, se pojavlja tudi na tem področju, le v bistveno manjšem obsegu.

Kot je mogoče že iz naslova diplomskega dela razbrati, je glavni problem odkriti plagiatstvo v programski kodi in prostem tekstu. Na internetu je že velika izbira programov in storitev, ki nam pomagajo pri takem odkrivanju. Vendar se že po prvih testiranjih izkaže, da niso vsi programi in storitve enako učinkoviti. Pred začetkom pisanja, sem si zastavil spisek programov in storitev, ki bi jih rad vključil v diplomsko delo. Ko sem pričel s pridobivanjem le-teh, sem kmalu prišel do problema, saj so bili programi, ki sem jih želel vključiti v diplomsko delo, plačljivi. Pisal sem elektronska sporočila skrbnikom programov in obrazložil situacijo, v kateri sem se znašel, vendar mi podjetja večinoma niso odobrila licence za programsko opremo. Več sreče sem imel pri pridobivanju orodij za odkrivanje plagiatstva v programski kodi, saj so mi avtorji dovolili testiranje njihovih orodij.

Za odkrivanje plagiatstva v prostem tekstu sem uporabljal zgolj brezplačne ali preizkusne različice storitev in programov. Pri tem sem podal svoje mnenje o storitvi ali programu, priložil zgled grafičnega vmesnika ter opisal najpomembnejše funkcije, ki jih imamo na voljo. Na koncu sem v tabeli prikazal, katera storitev ali program je odkrila večji odstotek plagiatstva v prostem tekstu. Za izvajanje le-teh sem uporabljal operacijski sistem Windows.

Odkrivanje plagiatstva v programski kodi sem izvajal v najbolj priljubljenih orodjih (MOSS, SIM, JPlag in Marble), ki jih še danes uporabljajo za testiranja na priznanih univerzah po vsem svetu. Opisal sem uporabo orodij, priložil zgled grafičnega vmesnika ter podal mnenje o učinkovitosti ali pomanjkljivostih. Nekatera orodja lahko izvajamo le na operacijskem sistemu Windows, za ostala pa sem uporabljal sistem Linux.

2 Plagiatorstvo

2.1 Zgodovina plagiatorstva

Do sedemnajstega stoletja pisatelji niso bili mnenja, da si prisvajajo avtorsko delo ali temo drugega avtorja, zato so kopiranje oziroma prepisovanje dojemali kot povsem moralno ali dovoljeno početje [42, 43]. Na tak način so posnemali velike mojstre tistega časa, se tako izobraževali in zgledovali po njih.

Že v četrtem stoletju pred našim štetjem je Aristotel napisal: »Imitacija je človeku naravna od otroštva, saj vse, česar se navadi, pridobi skozi proces imitacije.« Kar bi pomenilo, da je imitacija povsem naravna, saj človek že od otroštva pridobiva znanje s posnemanjem drugih ljudi.

V prvem stoletju se je pojavila uporaba latinske besede "plagiarius" (plagiator), ki še danes označuje nekoga, ki krade delo nekoga drugega. Prvič se beseda omenja, ko se je rimski pesnik Martial pritožil, da mu je drugi pesnik ukradel njegove verze.

Od konca osemnajstega stoletja naprej se v evropski kulturi pojavi zahteva, da so intelektualni proizvodi izvirni oziroma inovativni. Od takrat naprej se tudi plagiatorstvo obravnava kot kaznivo dejanje, saj je izvirnost dela zelo pomembna. Dramatik Ben Johnson je bil prvi, ki je besedo "plagiat" vnesel v angleški pravopis in obrazložil njen pomen.

Ameriški pisatelj, esejist in kritik Thomas Mallon, ki je raziskoval in objavil tudi nekaj literature o plagiatorstvu, je zatrdil, da je do takrat prevladoval klasičen pogled na prepisovanje, torej brez kakršnih koli sankcij. Zapisal je še, da je plagiatorstvo postalo problem v sedemnajstem stoletju, ko je pisanje postalo poklic. To je bila prelomnica, saj je pisatelj z izdajo literarnega dela lahko pričakoval, da bo kakšen bralec ali drug pisatelj posnemal kakšne njegove besede ali stavke. Tako je kopiranje besed izredno spremenilo vlogo pisatelja v družbi. Pred tem je bilo plagiatorstvo bolj stvar zabave kot pravnih sporov. Nenadoma postane avtor zelo cenjen v primerjavi s pripovedovalcem zgodb ravno obratno kot prej, saj je družba začela ceniti predvsem izvirnost literarnega dela.

2.2 Vrste in posledice plagiatorstva

2.2.1 Plagiatorstvo na splošno

Kot sem že v uvodu omenil, je področij, kjer se pojavlja plagiatorstvo, izredno veliko, saj ga najdemo v glasbi, znanosti, književnosti, novinarstvu, šolstvu, itd.

V glasbi pride največkrat do suma, da gre za plagiatorstvo, ko drug avtor posnema ali malenkost spremeni melodijo skladbe. Da gre za plagiat zelo težko dokažemo, saj se umetniki še danes zgledujejo drug po drugem. Ali je tudi besedilo pesmi plagiat, pa lahko preverimo s programi, ki odkrivajo plagiatorstvo v prostem tekstu.

V znanosti je večina plagiatorjev takih, ki bi radi čim več zaslužili ter pridobili ugled brez lastnega truda. Obstajajo pa tudi taki, ki ukradejo tuje avtorsko delo zaradi pomanjkanja svojega znanja in sposobnosti.

Književnost je že od nekdaj bila povezana s plagiatorstvom. Tukaj ni mišljeno zgolj, da bi si drug pisatelj prilastil odlomek drugega dela ali celotno delo, temveč je plagiatorstvo tudi kraja idej. Če je avtor v delu zapisal svojo misel ali idejo, je drugi avtor ne bi smel povzeti brez citiranja ali pa jo naprej predstavljati kot svoje osebno mišljenje.

Zaradi vse večjega razpolaganja z informacijami, ki nam jih ponuja novinarstvo, se plagiatorstvo širi tudi v tej dejavnosti. Sem štejemo zlasti prispevke, ki jih objavljajo v množičnih medijih. Ker imamo v Sloveniji nacionalno televizijsko hišo ter več komercialnih televizijskih postaj, se včasih zgodi, da pri komentiranju istega dogodka naletimo na iste ali zgolj obrnjene besede.

Vse več plagiatorstva je tudi v šolstvu, predvsem pri domačih, seminarskih in diplomskih nalogah, pa tudi v magistrskih delih in celo v doktorskih disertacijah. Na tem področju se plagiatorstvo odkriva predvsem zaradi čedalje bolj množičnega vpisa na študij in zaradi vse bolj izpopolnjene tehnologije, ki jo imamo na razpolago. O plagiatorstvu v šolstvu sledi podrobnejši opis v naslednjem podpoglavju.

Zadnja leta smo priča tudi raznim plagiatorskim aferam v politiki. Žalostno je, da imajo taki ljudje pomembne funkcije v državi, čeprav so do izobrazbe prišli na nezakonit način.

2.2.2 Plagiatorstvo v šolstvu

Vse več študentov uporablja internet za pridobitev zelenih analiz, interpretacij ali celo rešenih celotnih nalog, ki jih kasneje oddajo kot svoje lastno delo [2]. Zaradi takega početja študent ob končanju šolanja ne usvoji vsega potrebnega znanja in ima lahko kasneje težave pri nadaljnjem šolanju ali na delovnem mestu.

Zaradi povečevanja plagiatorstva, ki je prisotno na področjih izobraževanja, raziskovanja, razvoja programske opreme, itd., so na šolah pričeli s prenovo obstoječih in razvojem novih sistemov za odkrivanje plagiatov.

Strokovnjak s področja plagiatorstva M. Schiller [5] je zapisal, da obstajajo tri različne oblike plagiatorstva. Prva je preprosto kopiranje »beseda za besedo« iz revije, knjige, spletne strani, ne da bi kopirano besedilo citirali ali vsaj zapisali pravega avtorja. Druga oblika je plagiatorstvo, kjer se stavke drugače zapiše, kot so bili v izvirnem besedilu. Sem štejemo tudi dobesedne prevode v druge jezike, kjer se pisec ne zahvali avtorju. Zahvala se zahteva tudi v primeru, ko pisec spremeni nekaj besed, izpusti nekaj stavkov ali enako vsebino napiše na drugačen način. Tretja oblika temelji na uporabi idej drugih avtorjev. Tukaj govorimo o plagiatu, če pisec ne predloži originalne misli in zahvale izvornemu avtorju.

Največ razlogov za medsebojno prepisovanje med študenti se skriva pri (učno) slabših študentih. Ti velikokrat sodelujejo z drugimi študenti, ki so boljši na tem področju in naredijo večino potrebnega dela. Slabši študenti pa na koncu zgolj prepisejo rezultate v prepričanju, da so tudi oni usvojili določeno znanje. Veliko je tudi primerov, ko slabši študenti kopirajo in popravijo delo drugega študenta, največkrat brez njegovega dovoljenja in v upanju, da bo goljufanje ostalo neopaženo. Nekaj pa je tudi študentov, ki so zelo slabo motivirani in kopirajo delo drugega študenta, zgolj zato, ker se ne želijo potruditi.

2.3 Kaj izključuje plagiatorstvo?

V tem poglavju bi rad omenil, da poznamo posebne primere, kjer se sum plagiatorstva izključuje. To je v primerih, če je delo [6]:

- samoplgiat,
- pri dvojni stvaritvi,
- ko gre za nezavedno prevzemanje ter
- ko gre za inspiracijo ali parodijo.

2.3.1 Samoplgiatorstvo

Pri samoplgiatorstvu gre za ponovno uporabo svojega prejšnjega dela oziroma že prej objavljenega besedila (lahko je tudi v skrajšani obliki) ampak brez navedbe predhodne objave. Temu bi z drugimi besedami lahko rekli samocitiranje ali povzemanje lastnega dela, s tujko pa temu pravimo recikliranje. To lahko storimo, ne da bi uporabili pravila in pogoje citiranja. Pravno gledano s tem ni nič narobe, ampak v akademskih in raziskovalnih sferah je to povsem neetično, saj mora biti znanstveno in raziskovalno delo izvirno v vseh segmentih, ki niso citirani. Enako velja za novinarstvo, recimo za objave v časopisih, revijah ...

2.3.2 Dvojna stvaritev

Dvojna stvaritev pomeni, da je avtor ustvaril svoje lastno delo, obenem pa je drugi avtor ustvaril popolnoma enako delo. Da bi dva različna avtorja, ki se ne poznata in brez kakršnega koli sodelovanja ustvarila enako delo, je skoraj nemogoče. Zato je kasnejše izdano delo po vsej verjetnosti plgiat. Vendar lahko pride tudi do takega pojava pri ljudeh iz enakih področij; grafično oblikovanje, prevodi iz tujega jezika, predstavitve iste kulturne dediščine itd.

2.3.3 Nezavedno prevzemanje

Pri nezavednem prevzemanju gre za podzavestno spominjanje tujih del, ki jih združimo v naše avtorsko delo. Človek, ki veliko bere ali se na kakšen drug način izobražuje, lahko marsikaj zanimivega tudi nezavedno povzame. Tukaj mislimo predvsem na ideje, misli, stavke in verze drugega avtorja. Pri nezavednem prevzemanju avtorju manjka ustrezna krivdna oblika, zato delo ni plgiat. Kljub temu pa se objektivno še vedno šteje za kršitev avtorskih pravic.

2.3.4 Inspiracija ali parodija

Pojem inspiracija pomeni navdih, domišljijo, ki jo vzbuja pojav ali stvar. To bi pomenilo, da nas delo drugega avtorja spodbudi ter v nas sproži ustvarjalni proces za novo delo, ki ni reprodukcija ali predelava izvirnega dela. V umetnosti bi to pomenilo, da npr. vidimo čudovito sliko in v trenutku dobimo fantastično inspiracijo ter še sami poskušamo narediti povsem enako stvar. Podobno bi to veljalo tudi pri literaturi, ko povzamemo enak naslov nato pa začnemo to poglavje opisovati. Stvari še zdaleč ne bodo enake, si bodo pa lahko zelo podobne.

Parodija pa je delo, ki na posmehljiv oziroma porogljiv način posnema zlasti vsebino drugega, resnega dela. Tako delo postane vsakomur razpoznavno in se v tem razlikuje od plagiata. Obstaja posebna pravna podlaga za dovoljenost parodije, ki pa zahteva, da upoštevamo splošne pogoje nekaterih členov ZASP (Zakon o avtorskih in sorodnih pravicah).

2.4 Kazenskopravni vidiki plagiata

Podobno kot v drugih državah tudi v slovenskem ZASP ne najdemo definicije plagiata in njegovih zakonskih znakov. Tako smo pri tem ključnem vprašanju vezani na tujo ali domačo doktrino, ki pa glede tega ni enotna. Veljavno avtorsko pravo ima oblikovana pravila glede varovanja avtorskih del, glede vsebine in pravic teh del, glede sankcij itd. Če pa je katero od teh pravil kršeno, imenujemo to kršitev avtorskih pravic. Za takšno dejansko stanje ne potrebujemo pojma plagiata, saj je v avtorskem pravu nepotreben [41].

Plagiat ali nekateri njegovi znaki se s pravnega vidika obravnavajo predvsem v okviru prava znanosti, avtorskega prava in kazenskega prava. Kot sem že omenil avtorsko pravo ne pozna zakonske definicije plagiata, različno ga opredeljuje tudi doktrina. Kazensko pravo pa se najbolj približuje pojmu plagiat, ko sankcionira kršitev moralne avtorske pravice [6].

Če se očitki, da je neko delo plagiat, izkažejo za utemeljene, v večini primerov doleti takega avtorja zgolj moralna obsodba ter s tem povezane sankcije, ki so določene z avtonomnimi pravili znanstvene institucije ali profesionalnega združenja. V primeru, da se tako početje ugotovi v izobraževalnih ustanovah, zlasti v primerih diplomskih, magistrskih ali doktorskih del, pa sledi odvzem pridobljenega naziva. Če je bil naziv pridobljen v tuji državi, se njegova veljavnost razveljavi, ter tako ostane brez pridobljenega naziva zunaj države, kjer je plagiator študiral [3].

»Seveda so sankcije, ki zadenejo plagiatorja, odvisne od tega, ali gre za poseg v avtorskoppravno zavarovano delo ali ne. Pri kršitvi avtorske pravice pridejo v poštev civilnopravne sankcije po ZASP. Ta v 167. členu ureja pravico do sodnega varstva, ki med drugim obsega prepoved kršitve in bodočega kršenja ter odpravo stanja, ki je nastalo s kršitvijo, prizadetemu pa priznava tudi pravico do povračila materialne škode in civilne kazni (168. člen ZASP) ter pravico do denarnega zadoščenja za nepremoženjsko škodo (168. člen ZASP). Kazenski zakonik (KZ-1) pa v zvezi s tem določa dve kaznivi dejanji: kršitev moralnih avtorskih pravic (147. člen) in kršitev materialnih pravic (148. člen)« ([3], str. 987).

V prvem odstavku 147. člena Kazenskega zakonika (KZ-1) je zapisano varovanje avtorskega dela pred klasičnim plagiatom, ki na neposreden način posega v pravico do priznanja avtorstva. To pravico krši vsak, ki s svojim imenom ali imenom koga drugega prikaže, izvede ali prenese tuje avtorsko delo oziroma del njegove literature. Najpogostejša oblika plagiata, predvsem pri pisanih delih, je dobesedno prepisovanje (prevzemanje) tujega dela z navedbo neresničnega lastnega avtorstva [4].

V 148. členu KZ-1 predpisane kazni omogočajo ustrezno sankcioniranje glede na težo kaznivega dejanja z vidika višine premoženjske vrednosti oziroma pridobljene premoženjske koristi. Znotraj tega lahko sodišče izbere in odmeri sankcijo v skladu s posebnostmi vsakega obravnavanega primera kršitve. V praksi pa se največkrat izreče zaporna ali pogojna zaporna kazen, izjemoma tudi denarna kazen. Take obravnave so pri nas zelo redke.

Tudi iz statistike lahko razberemo, da pri nas kazenskopravnega obravnavanja plagiata kot kršitve moralne avtorske pravice praktično ni [4].

Plagiator velikokrat poseže tudi v materialne avtorske pravice. Te so kršene toliko, kolikor je tujih stvaritev, ki jih je avtor neupravičeno povzegal ter je tako postalo predmet pridobivanja oziroma uveljavljanja takih pravic iz lastnega naslova, torej z njegovo nedovoljeno objavo, prikazovanjem, izvajanjem ali prenosom. Objektivni pogoj kaznivosti je, da skupna tržna cena neupravičeno uporabljenih avtorskih del oziroma njihovih primerkov doseže večje premoženjsko vrednost, saj mora presežati 5.000 evrov. Pri kršitvi materialnih avtorskih pravic s plagiatom je še posebej pomembna vrednost prenesene avtorske vsebine in s tem obseg pridobitnega namena ter dokazovanja.

2.5 Kako omejiti plagiatorstvo?

Plagiatorstvu se lahko izognemo tako, da smo vedno zelo pozorni pri navajanju (citiranju, ustnem poročanju ali povzemanju) virov, kadar dobesečno navajamo besede nekoga drugega [26, 28, 38, 39]. Ravno tako velja za povzemanje besed, ki se navezujejo na delo drugega avtorja, razen ko gre za znane in splošno sprejete ideje. Pa tudi pri reproduciranju drugih vrst dela, kot so tabele, grafi, slike, fotografije ...

Če se želimo izogniti nehotenemu plagiatu, si najlažje pomagamo tako, da že med pisanjem dela natančno zabeležimo izvor virov in ko ustvarjamo zapiske za te besede, uporabimo narekovaje prav povsod, kjer so dobesečno prepisane iz nekega drugega dela, tudi če gre le za del stavka ali besedne zveze.

Izredni profesor Milan Ojsteršek, ki poučuje na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru, je priznani strokovnjak na področju računalniško podprtega odkrivanja plagiatorstva. Dejal je, da preverjanje plagiatorstva ne bo nikoli popolno. Razlog je predvsem ta, da sistemi za odkrivanje plagiatorstva ne sledijo spretnim goljufom, saj ti uporabljajo najnovejše trike, da prikrijejo svoj ponaredek [30].

Profesor Ojsteršek je tudi vodja digitalne knjižnice oziroma repozitorija, ki ga od leta 2008 uporablja Univerza v Mariboru. V digitalni knjižnici so shranjena diplomska dela, magisteriji, doktorati ter znanstveni časopisi in zbirke Univerze [11]. Preden se literatura odda v digitalno zbirko, jo pregledajo s programi za odkrivanje plagiatorstva. Če plagiatorstva ni, se literatura doda v bazo podatkov v nasprotnem primeru pa ne.

Podoben repozitorij ima tudi Univerza v Ljubljani, ki omogoča zbiranje in hranjenje elektronskih oblik zaključnih del študija, objav zaposlenih na Univerzi ter raziskovalnih podatkov [33]. Tudi tukaj se v procesu oddaje izvede preverjanje enakosti (plagiatorstva). Zato lahko z gotovostjo trdimo, da so gradiva, ki jih hranijo digitalne knjižnice oziroma repozitoriji, zagotovo avtorska dela.

Res pa je, da se pri ugotavljanju podobnosti ni mogoče v celoti zanesti na različne računalniške programe za odkrivanje plagiatov, saj ti programi razpolagajo z omejenim številom podatkovnih baz, ki ne vsebujejo vseh besedil. Pri tem lahko gre tudi za prevod tujega besedila v domač jezik, česar programi za odkrivanje plagiatorstva ne ugotovijo. Lahko pa se uporabljajo tudi različni triki, npr. da želeno besedilo razdelimo na več kratkih stavkov ali več stavkov združimo v enega ter tako prelisichimo program za odkrivanje plagiatorstva.

3 Avtorske pravice

Plagiatorstvo je kršitev avtorske pravice, saj je ta po ZASP enovita pravica na avtorskem delu, ki avtorju zagotavlja uresničevanje premoženjskih (materialnih) in osebnih (moralnih) interesov v zvezi z izkoriščanjem avtorskega dela.

Moralne pravice se nanašajo na izključna osebna upravičenja avtorja in varujejo njegove duhovne in osebne vezi do dela. Prav tako je tukaj določena pravica do prve objave, priznanja avtorstva, spoštovanja dela itd.

Materialne pravice so namenjene temu, da avtorju omogočijo uživanje vrednosti, ki jo je ustvaril s svojim delom. Varujejo izključna premoženjska upravičenja, določajo pravico reproduciranja, pravico javnega izvajanja, prenašanja, predvajanja, prikazovanja, oddajanja ... Poleg tega določajo tudi uporabo dela v spremenjeni obliki (pravica predelave, priredbe) ter uporabo primerkov avtorskega dela (pravica distribuiranja, dajanja v najem).

Avtorske pravice (copyright) [16, 18] so skupek pravnih pravil, s katerimi se urejajo družbeni odnosi glede intelektualnega ustvarjanja. Ustava Republike Slovenije v 60. členu določa, da je zagotovljeno varstvo avtorskih in drugih pravic, ki izvirajo iz umetniške, znanstvene, raziskovalne in izumiteljske dejavnosti.

Avtorska pravica nastane s samo stvaritvijo dela. Poseben postopek registracije ni potreben, vendar je smiselno avtorsko delo vpisati v register avtorskih del, ki ga vodi Avtorska agencija za Slovenijo, ker se šteje, da na tem delu obstajajo avtorske pravice in da pripadajo osebi, ki je v registru označena kot imetnik. Njegove pravice trajajo ves čas avtorskega življenja in 70 let po njegovi smrti.

Avtorske pravice se lahko prenašajo samo s pisno pogodbo. Pravice uporabnikov se urejajo z licencami. Kolektivno upravljanje avtorskih pravic omogoča avtorjem in drugim imetnikom pravic preprostejšo in učinkovitejšo upravljanje pravic v razmerah množičnega koriščenja njihovih del, uporabnikom pa enostaven dostop do zakonite uporabe številnih del.

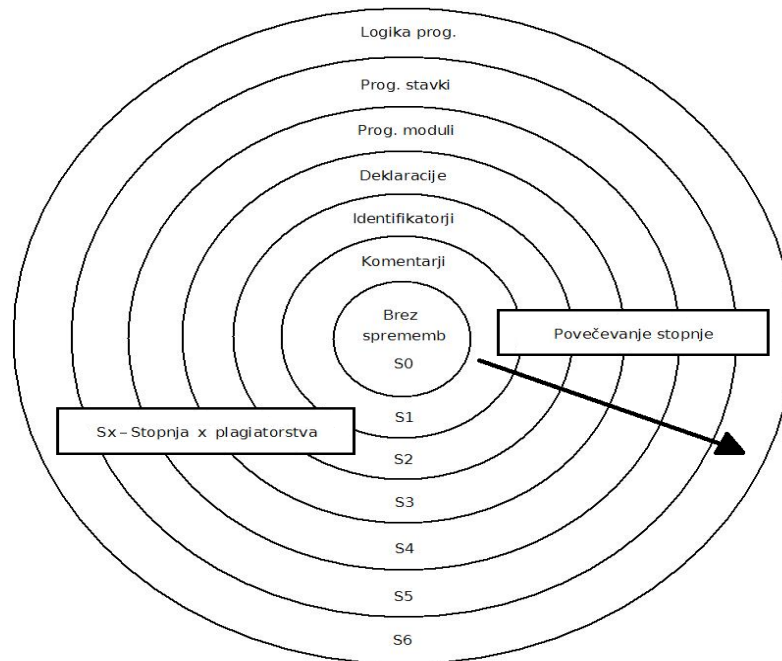
Dovoljenje za kolektivno uveljavljanje oziroma upravljanje avtorskih pravic v Republiki Sloveniji imajo naslednje večje organizacije: Združenje SAZAS, Društvo ZAMP, Zavoda IPF in AIPA ...

4 Odkrivanje plagiatorstva v programski kodi

Tako kot se pojavlja plagiatorstvo pri domačih, seminarskih, diplomskih ter ostalih nalogah, se pojavlja tudi v programih. Plagiatorstvo programske opreme, ki sledi miselnosti raziskovalcema algoritmov za odkrivanje plagiatorstva v izvorni kodi Parkerja in Hamblena, definira plagiarizirani program kot: »Program, ki izhaja iz drugega programa, kot rezultat manjšega števila operacij tekstovnega urejanja, vendar (brez) natančnega razumevanja programa« [1]. Te spremembe so lahko drugačni komentarji, druge spremenljivke, spremenjen grafični vmesnik, drugačna odločitvena logika programa ...

Plagiatorstvo programske opreme na področju študija v računalništvu je prisotno najmanj od leta 1970. Leta 1987 sta J. A. W. Faidhi in S. K. Robinson [6] na Univerzi Brunel objavila knjigo z naslovom "Empirični pristop za odkrivanje programskih podobnosti in plagiarizma v univerzitetnem programskem okolju". Faidhi in Robinson definirata šest stopenj v spremembah izvorne kode, ki izhaja iz programa plagiatorstva, kot je prikazano na Sliki 1. Notranji krog je na najosnovnejši ravni in predstavlja kategorijo programov, kjer ni spremembe v izvorni kodi. Bolj kot se pomikamo proti zunanosti, več kompleksnih sprememb se pojavi. Prva stopnja vključuje spremembo komentarjev in zamikov. Na drugi stopnji so vključene spremembe na prvi stopnji ter spremembe identifikatorjev (spremenjena imena spremenljivk, konstant, procedur, funkcij ali spremenjeno ime programa). Stopnja tri vključuje spremembe v stopnji dve in spremembe deklaracij (deklaracija dodatnih spremenljivk in konstant, zamenjan vrstni red procedur in funkcij itd.). Na četrti stopnji so vključene vse spremembe na tretji stopnji in spremembe na programskih modulih (primer: razdelitev procedure na več procedur, združevanje več procedur v eno, kreiranje novih procedur in funkcij itd.). Peta stopnja vključuje spremembe iz četrte stopnje in spreminjanje programskih stavkov (primer: zanko for spremenimo v while, zanko while spremenimo v repeat itd.). Stopnja šest vključuje vse spremembe na stopnji pet vključno s spremembami v odločitveni logiki programa (primer: spremembe v izrazih).

Lahko pa imamo tudi primer, ko pisec ne napravi nobene spremembe, vendar se tak plagiat precej hitro odkrije. Razvrščanje po stopnjah je uporabno, vendar dokaj poljubno. Kasneje so se v literaturi pojavile še druge razvrstitvene možnosti.



Slika 1: Stopnje plagiatorstva v programski kodi [15].

4.1 Orodja za detekcijo plagiatorstva v programski kodi

Ker je bilo ročno odkrivanje enakosti v programski kodi časovno zelo potratno, so stremeli k temu, da bi preverjanje postalo avtomatizirano. Prvi tak sistem je prišel na trg leta 1976. Od takrat naprej je nastalo že veliko število orodij za odkrivanje podobnosti v izvorni kodi. Orodja [17, 21, 22, 23] delujejo tako, da primerjajo dve ali več datotek izvorne kode med seboj ter izpišejo, koliko procentov programske kode je vsebovane v drugih datotekah. Če je odstotek vsebovanosti zelo majhen, po vsej verjetnosti ne gre za plagiat. Zato imamo ponavadi možnost, da že pred začetkom preverjanja nastavimo, kolikšen procent kode mora biti enak, da se dokument nadalje obravnava. Na voljo imamo brezplačne programe, odprtokodno programsko opremo in plačljive (komercialne) programe.

Za odkrivanje plagiatorstva v programski kodi največkrat uporabljamo orodja MOSS, JPlag, SIM, Marble in Plaggie. V nadaljevanju podrobneje opisal orodje MOSS, JPlag, SIM in Marble.

4.1.1 MOSS

MOSS (Measure Of Software Similarity) je sistem [35] za določanje podobnosti programske kode. Razvil ga je Alex Aiken leta 1994 na Univerzi v Stanfordu za odkrivanje plagiatorstva v programskih razredih. Od takrat je eden največjih in zelo učinkovitih sistemov. Algoritem po katerem deluje sistem MOSS, je bolj izpopolnjen v primerjavi z algoritmi, ki jih uporabljajo druga orodja za odkrivanje plagiatorstva.

Trenutno lahko analiziramo kodo iz naslednjih programskih jezikov: C, C++, Java, C#, Python, Visual Basic, Javascript, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Matlab, VHDL, Verilog, Spice, MIPS assembly, a8086 assembly, a8086 assembly, MIPS assembly in HCL2.

Sistem MOSS [9] deluje kot spletna storitev, zato so algoritmi, ki jih sistem uporablja za delovanje uporabniku nedostopni. Na medmrežju se je pojavil članek, v katerem so avtorji sistema nadzorno opisali logiko sistema, na kateri temelji. Za to se uporablja algoritem robustnega presejanja (winnowing), ki uporablja za odkrivanje plagiatorstva pristop digitalnih prstnih odtisov (digital fingerprint). Princip digitalnih prstnih odtisov je izbor določene podmnožice iz rezultatov, ki jih vrne zgoščevalna funkcija ob pregledu dokumenta. Poteka tako, da se besedilu v dokumentih odstranijo nepomembne značilnosti, kot so velike začetnice, presledke, ločila itd. Novonastalo besedilo se nato razdeli na posamezne besede dolžine k (k -gram), kjer je k konstanta (določi jo uporabnik). Vsakokrat se pomaknemo za eno mesto naprej ter vzamemo novo dolžino besedila k -gramov in to ponavljamo vse do konca besedila. Ti k -grami se nato z pomočjo zgoščevalnih funkcij pretvorijo v vrednosti. Iz te množice vrednosti se nato z enim od algoritmov (npr. z naključnim deljenjem z ostankom) izberejo vrednosti, ki se bodo preverjale z vrednostmi v ostalih dokumentih in odkrivala podvajanja. Ta podmnožica predstavlja prstne odtise dokumenta.

Algoritem robustnega presejanja je bolj učinkovit in razširljiv, saj izbere manj prstnih odtisov za dosego iste kakovosti rezultatov, kot to počno ostali algoritmi. Ta iz podmnožice prstnih odtisov vzame število w (določa širino okna), ki ga določi uporabnik. Celotno zaporedje vrednosti pridobljenih s pomočjo zgoščevalnih funkcij se nato zapiše v zaporedje oken (širine w). Nato iz vsakega okna izberemo najmanjše število v zaporedju. Končna urejena zaporedja pa bodo predstavljala prstni odtis dokumenta.

Za namene te aplikacije se shrani z vsakim izbranim prstnim odtisom v digitalni prstni odtis še številka dokumenta in vrstice. V prvem koraku se ustvari seznam oziroma kazalo prstnih odtisov, ki priredi prstne odtise do lokacij za vse dokumente – podobno kot invertni indeks, ki ga ustvarijo iskalniki (priredijo besede do mest v najdenih dokumentih). V drugem koraku se

vsakemu dokumentu priredi prstne odtise še drugič, izbrane prstne odtise pa nato najde na seznamu, ki je bil zgrajen predhodno. Operacija poda seznam vseh ujemaajočih se prstnih odtisov za vsak dokument. Potemtakem seznam ujemaajočih se prstnih odtisov za dokument d lahko vsebuje prstne odtise iz mnogih različnih dokumentov d1, d2 itd. V naslednjem koraku se seznam ujemaajočih prstnih odtisov za vsak dokument d razvrsti glede na dokument in ujemanja za vsak par dokumentov (d, d1), (d, d2). Ujemanja med dokumenti se uredijo glede na velikost (število prstnih odtisov) in največja ujemanja se prikažejo uporabniku. Pri tem je pomembno, da ujemanja ne padejo pod uporabniško nastavljen prag.

MOSS je računalniško orodje, ki deluje na Windows ali Unix/Linux operacijskih sistemih. Sam sem uporabljal verzijo, ki deluje na Linux sistemu. Preden zaženemo sistem, mu določimo, katere parametre naj upošteva (kakšen programski jezik bomo preverjali ter določimo seznam datotek, ki naj se preverjajo med seboj). Te datoteke se nato prenesejo na MOSS spletni strežnik (se nahaja na Univerzi v Stanfordu), kjer se primerjajo med seboj. Pri tem se zanemarijo tudi razne knjižnice ali vnaprej podana koda, za katero menimo, da je uporabljena v več dokumentih, s čimer se odpravi lažne alarme suma plagiatorstva. Če je primerjanje datotek uspelo, nam strežnik vrne rezultate v obliki HTML dokumentov. Do teh dokumentov lahko dostopamo tudi preko URL povezave, ki se nam prikaže v terminalu. V nasprotnem primeru nam izpiše napako.

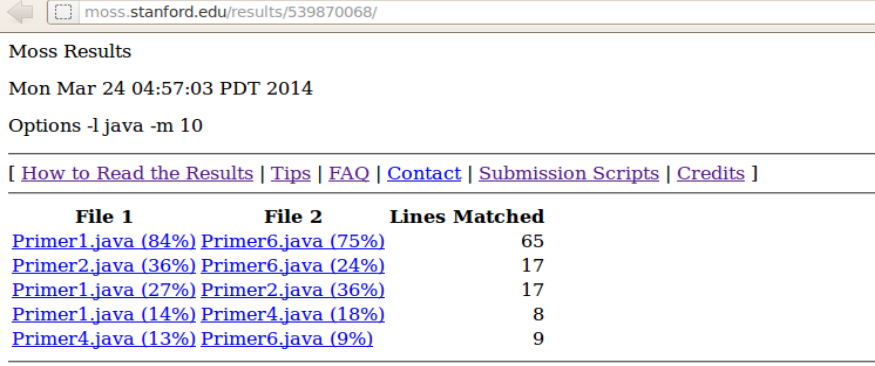
S klikom na URL povezavo dostopamo do predstavitve rezultatov na HTML spletni strani, ki se je ustvarila, ko je sistem MOSS končalo preverjanje dokumentov. Glede varnosti je poskrbljeno, saj so rezultati na spletni strani prikazani zasebno. Do njih ne moremo dostopati s pomočjo iskalnih algoritmov ali preprosto z brskanjem po internetu. URL povezava do spletne strani pa ni shranjena neomejeno, saj preneha veljati že po dveh tednih. Tako lahko predvidevamo, da je potencial za zlorabo informacij precej majhen.

V odgovoru na poizvedbe strežnik MOSS proizvede URL povezavo na HTML dokumente, ki nam prikažejo rezultate v obliki HTML spletne strani (Slika 2).

```
anej@anej-desktop:~$ ./moss -l java Primer1.java Primer2.java Primer3.java Primer4.java Primer5.java Primer6.java
Checking files . . .
OK
Uploading Primer1.java ...done.
Uploading Primer2.java ...done.
Uploading Primer3.java ...done.
Uploading Primer4.java ...done.
Uploading Primer5.java ...done.
Uploading Primer6.java ...done.
Query submitted. Waiting for the server's response.
http://moss.stanford.edu/results/539870968
anej@anej-desktop:~$
```

Slika 2: Poganjanje sistema MOSS.

Predstavitev rezultatov je torej HTML prikaz, kjer so prikazani pari datotek, ki imajo podobno kodo ter so razvrščeni po vsebovanosti enakosti kode med seboj (Slika 3). Vedno se primerjata po dve datoteki med seboj in na koncu se izpiše število vrstic programske kode, ki se ujemajo. Pri vsaki datoteki je naveden tudi odstotek v procentih, ki prikaže, kolikšen delež enake kode se ujema s kodo v drugi datoteki v odvisnosti od skupne velikosti programske kode v datoteki. Višja kot je številka, več je enakosti ali plagiatorstva.



Moss Results
Mon Mar 24 04:57:03 PDT 2014
Options -l.java -m 10

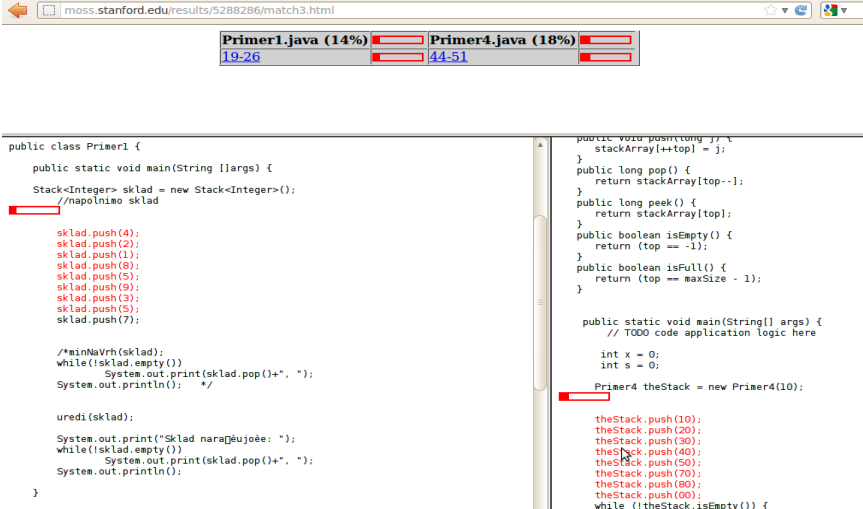
[[How to Read the Results](#) | [Tips](#) | [FAQ](#) | [Contact](#) | [Submission Scripts](#) | [Credits](#)]

File 1	File 2	Lines Matched
Primer1.java (84%)	Primer6.java (75%)	65
Primer2.java (36%)	Primer6.java (24%)	17
Primer1.java (27%)	Primer2.java (36%)	17
Primer1.java (14%)	Primer4.java (18%)	8
Primer4.java (13%)	Primer6.java (9%)	9

Any errors encountered during this query are listed below.

Slika 3: MOSS prikaz rezultatov.

S klikom na posamezno vrstico v tabeli, ki prikazuje rezultate primerjanja dveh datotek, nas spletna povezava postavi na novo spletno stran. Tukaj je grafično prikazano, katera koda se ponavlja v obeh datotekah (Slika 4). Na vrhu nam prikaže, katere vrstice v prvi datoteki se ujemajo z enakim zapisom v drugi datoteki. Spodaj pa imamo prikazano celotno programsko kodo obeh datotek in kjer je besedilo enako, je označeno z rdečo barvo. Tako lahko najbolj nazorno vidimo, da se besedilo res pojavlja v obeh datotekah.



moss.stanford.edu/results/5288286/match3.html

Primer1.java (14%) **Primer4.java (18%)**

19-26 44-51

```

public class Primer1 {
    public static void main(String []args) {
        Stack<Integer> sklad = new Stack<Integer>();
        //napolnimo sklad
        sklad.push(4);
        sklad.push(2);
        sklad.push(1);
        sklad.push(9);
        sklad.push(5);
        sklad.push(9);
        sklad.push(3);
        sklad.push(5);
        sklad.push(7);

        /*mainNavrh(sklad):
        while(!sklad.empty())
            System.out.println(sklad.pop()+" ");
        System.out.println(); */

        uredi(sklad);
        System.out.println("Sklad nara\u0161uje: ");
        while(!sklad.empty())
            System.out.println(sklad.pop()+" ");
        System.out.println();
    }
}

public class Primer4 {
    private static int top;
    private static Integer[] stackArray;

    public long pop() {
        return stackArray[top--];
    }
    public long peek() {
        return stackArray[top];
    }
    public boolean isEmpty() {
        return (top == -1);
    }
    public boolean isFull() {
        return (top == maxSize - 1);
    }

    public static void main(String[] args) {
        // TODO code application logic here
        int x = 0;
        int s = 0;

        Primer4 theStack = new Primer4(10);

        theStack.push(10);
        theStack.push(20);
        theStack.push(30);
        theStack.push(40);
        theStack.push(50);
        theStack.push(70);
        theStack.push(80);
        theStack.push(90);
        while (!theStack.isEmpty()) {

```

Slika 4: MOSS podrobnejši prikaz podobnosti.

Rezultata, ki nam ga sistem MOSS izpiše, ni mogoče obravnavati kot neposreden dokaz za plagiatorstvo, saj potrebujemo še nekoga, ki bo preučil program ter izrekel končno odločitev.

MOSS je brezplačen sistem, vendar je pred uporabo potrebna registracija uporabnika. Ni odprtokodni sistem, kar pomeni, da nimamo dostopa do razvojne metodologije ali programske kode produkta.

4.1.2 SIM

Sim [25] je še eden od sistemov za odkrivanje podobnosti v programski kodi. Razvil ga je Dick Grune leta 1989 na Vrije Universiteit v Amsterdamu. Trenutna različica orodja za operacijski sistem Windows je 2.77, ki je bila izdana na začetku leta 2014. Sistem poganjamo v ukazni lupini (cmd). Na voljo pa imamo tudi različico za poganjanje v Unix (Linux) operacijskih sistemih, s podajanjem ukazov v lupino operacijskega sistema. Z njim ugotavljamo plagiatorstvo v programskih jezikih C, Java, Pascal, Modula – 2, Lisp, Miranda in tekstovnih datotekah.

Sistem SIM se uporablja za odkrivanje podobnosti v programski kodi, ki je bila pridobljena na nepošten način. Najprej ustvari referenčno tabelo, ki se lahko uporablja za odkrivanje na novo vpisanih datotek in besedil, ki jih želimo preverjati.

Medtem ko sistem SIM ne zagotavlja več aktivnega vzdrževanja in podpore, je njegova izvorna koda na voljo tudi javnosti. Glede na potek razvoja in na avtorje, ki so sodelovali pri tem projektu, je sistem SIM zelo prilagojeno uporabi na Vrije Universiteit v Amsterdamu in zato ni zelo razširjen in popularen.

Algoritem [8], ki deluje v sistemu SIM, pretvori izvorno kodo v nize žetonov in nato primerja te nize poravnava z uporabo tehnike dinamičnega programiranja nizov. Ta tehnika se uporablja tudi v nizu DNA ujemanja, ki pripomore k zaporedni poravnavi nizov. Poravnava je zelo potratna za izračunavanje, zato SIM ni prilagodljiv, saj ne omogoča velikega skladiščenja podatkov.

Sam sem poganjal sistem SIM v Windows komandni vrstici, ki služi kot vmesnik med uporabnikom in jedrom operacijskega sistema. Ukaz pričnemo z imenom ukaza, ki mu po potrebi dodamo stikala in dodatne argumente (v našem primeru imena datotek). Za testiranje sem uporabljal ukaz za programski jezik Java »sim_java.exe«.

Spodnja Slika 5 prikazuje, kako lahko ustvarimo dvostolpčni izpis. Sistem pregleda in primerja programsko kodo datotek med seboj, če ugotovi enakost (ali zgolj deloma spremenjeno programsko kodo), nam to kodo izpiše. Izpisane so tudi vrstice v obeh datotekah, ki se ujemajo. Tako najlažje preverimo, ali res gre za plagiatorstvo.

```
C:\Users\Bandelj\Downloads\sim_exe_2_77>sim_java.exe -fF Primer1.java / Primer2.
java / Primer3.java / Primer4.java / Primer5.java / Primer6.java
File Primer1.java: 331 tokens, 99 lines
File /: separator
File Primer2.java: 238 tokens, 47 lines
File /: separator
File Primer3.java: 298 tokens, 68 lines
File /: separator
File Primer4.java: 257 tokens, 61 lines
File /: separator
File Primer5.java: 157 tokens, 52 lines
File /: separator
File Primer6.java: 374 tokens, 93 lines
Total: 1655 tokens

Primer2.java: line 12-21      |Primer6.java: line 11-19      | [45]
public class Primer2 {      |public class Primer6 {
public static void main(String []args) |public static void main(String []args)
Stack<Integer> sklad = new Stack<Integ |Stack<Integer> sklad = new Stack<Integ
sklad.push(6);              |//napolnimo sklad
sklad.push(0);              |sklad.push(0);
                              |sklad.push(9);
                              |sklad.push(?);
System.out.print("Sklad normalno izpi |
```

Slika 5: SIM dvostolpčni izpis podobnosti programske kode.

Slika 6 prikazuje, katera stikala moramo dodati ukazu, da bo sistem pregledal celotno programsko kodo v datotekah ter podal rezultat v procentih, koliko enakosti vsebujejo datoteke. Procent se izračuna glede na število ujemanj programske kode v odvisnosti od skupne velikosti programske kode v datoteki. Ali gre resnično za plagiatorstvo, pa mora preučiti človek.

```
C:\Users\Bandelj\Downloads\sim_exe_2_77>sim_java.exe -sp Primer2.java Primer4.j
ava Primer6.java Primer1.java Primer3.java Primer5.java
File Primer2.java: 213 tokens, 47 lines
File Primer4.java: 231 tokens, 61 lines
File Primer6.java: 345 tokens, 93 lines
File Primer1.java: 303 tokens, 99 lines
File Primer3.java: 266 tokens, 68 lines
File Primer5.java: 131 tokens, 52 lines
Total: 1489 tokens

Primer1.java consists for 99 % of Primer6.java material
Primer6.java consists for 92 % of Primer1.java material
Primer2.java consists for 36 % of Primer6.java material
Primer4.java consists for 22 % of Primer6.java material
```

Slika 6: SIM primerjava datotek med seboj.

Rezultate lahko prilagodimo tudi tako, da želene datoteke primerjamo s točno določeno datoteko. Tak postopek prikazuje Slika 7.

```

C:\Users\Bandelj\Downloads\sim_exe_2_77>sim_java.exe -epS Primer1.java / Primer3
.java / Primer4.java / Primer5.java / Primer6.java / Primer2.java
File Primer1.java: 303 tokens, 99 lines
File /: separator
File Primer3.java: 266 tokens, 68 lines
File /: separator
File Primer4.java: 231 tokens, 61 lines
File /: separator
File Primer5.java: 131 tokens, 52 lines
File /: separator
File Primer6.java: 345 tokens, 93 lines
File /: separator
File Primer2.java: 213 tokens, 47 lines
Total: 1489 tokens

Primer6.java consists for 66 % of Primer2.java material
Primer1.java consists for 55 % of Primer2.java material

```

Slika 7: SIM primerjava datotek z izbrano datoteko.

Vse prikaze rezultatov lahko shranimo tudi v datoteko. To storimo tako, da ukazu dodamo stikalo "-o ImeDatoteke" in ta nam kreira novo datoteko ter vanjo zapiše rezultate analize. Ta način je zelo uporaben zlasti pri večjih datotekah, saj se precej izboljša preglednost in rezultati so tako trajno hranjeni.

Ker je Sim odprtokodno orodje, ima možnost razširitve ali izboljšanja programa. Na ta način lahko dopolnimo orodje, ki nam bo najbolj ustrezalo v situaciji, ki jo nameravamo uporabiti.

4.1.3 JPlag

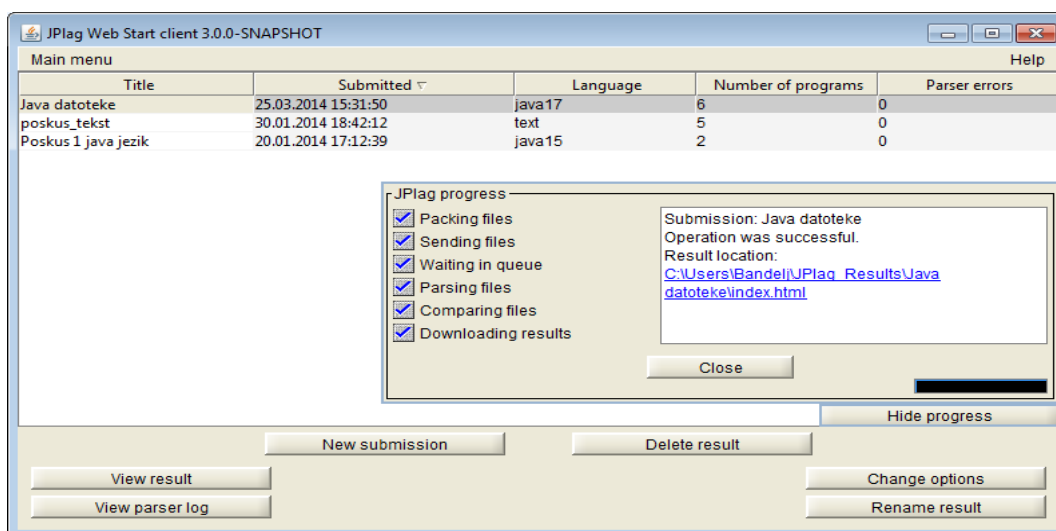
Je sistem [24], ki ga je razvil Guido Malphol na Univerzi v Karlsruhe. Ta primerja programsko kodo v več datotekah ter med njimi odkriva podobnosti. Na tak način se torej zazna plagiatorstvo v programski kodi. Sistem JPlag ne primerja zgolj bajtov besedila, ampak se pri preverjanju poglobi v sintakso programskega jezika in v programsko strukturo. Orodje je zelo podrobno in natančno pri odkrivanju plagiatorstva. JPlag trenutno podpira programski jezik Java, C#, C, C++, Scheme in iskanje plagiatorstva v tekstovnih datotekah.

JPlag se običajno uporablja za odkrivanje in preprečevanje nedovoljenega kopiranja programske kode med dijaki ali študenti v študijskih programih izobraževalnih ustanov. Vendar pa se lahko načeloma uporablja tudi za odkrivanje izvornega besedila med velikimi količinami teksta ter tako odkrijemo, kje je prišlo do podvajanja (ali le malenkostnega spreminjanja) besedila. Tako lahko orodje JPlag uspešno pripomore pri preprečevanju kraje intelektualne lastnine.

Tudi JPlag [7] deluje kot spletna storitev, zato so algoritmi, ki jih orodje uporablja za uporabnikov vpogled nedostopni. Imamo pa na voljo opis, ki nam predstavi delovanje algoritma za odkrivanje plagiatorstva. Njegova primerjava temelji na žetonih. Najprej se preberejo datoteke, ki jih želimo preverjati ter s pomočjo razčlenjevalca pretvorijo izvorno kodo v žetone. Nato se ti pari žetonov primerjajo med seboj, ugotavlja se njihova podobnost (podana v odstotkih). Pretvorbe vrstic kode v žetone so odvisne od programskega jezika, ki ga uporabljamo. Ustvarjeni žetoni ne vsebujejo komentarjev in presledkov. Če primerjamo dva niza A in B, je cilj najti nabor podnizov, ki so enaki in izpopolnjujejo naslednja pravila: vsak žeton iz niza A se lahko ujema z enim parametrom žetona iz niza B, podnize moramo najti neodvisno od njihovega položaja v nizu (prerazporeditev kode), dolgi podnizi ujemanj so zanesljivejši kot krajši, saj ti zagotavljajo manj možnosti za naključja. Algoritem Running Karp – Rabin Greedy String Tiling (RKR – RGST) pri delovanju poišče najdaljše nize, jih označi ter izvede pregled ujemanja žetonov. Postopoma s programskimi zankami zmanjšuje dolžine nizov in ugotavlja enakosti žetonov, dokler ne pride do minimalne dolžine, ki je določena za ujemanje. Te rezultate nato programsko obdela in pošlje uporabniku.

Preden pričnemo z uporabo, je potrebna namestitev aplikacije, ki deluje kot klient za uporabo njihovih spletnih servisov. Ko aplikacijo zaženemo, najprej na seznamu izberemo, katere vrste datotek (programski jezik ali tekstovno besedilo) bomo pregledovali (Slika 8). Nato napišemo, kako naj se v bazi poimenuje to preverjanje ter izberemo datoteke, ki jih bomo uporabljali. Lahko izberemo tudi celotno mapo. V tem primeru bodo v preverjanje vključene tudi vse podmape in datoteke, ki vsebujejo predhodno izbran programski jezik ali tekstovno besedilo.

Izbrane datoteke nato aplikacija klienta pripravi za pošiljanje na strežnik, kjer se postavijo v čakalno vrsto za pregled. Čas izvedbe je odvisen od zasedenosti strežnika. Ko se procesiranje prične, se izvede razčlenjevalec (parser), ki razčleni programsko kodo datotek v obliko, ki bo primerna za primerjanje med seboj. Primerjanje se izvrši na strežniku in ko je končano, nam strežnik vrne rezultate v obliki poročila, ki se shrani na našem trdem disku. Do teh rezultatov lahko dostopamo preko URL povezave na začetno HTML stran »index.html«, saj se ti rezultati hranijo v zapisu za spletne strani.



Slika 8: JPlag poganjanje klienta.

S klikom na to povezavo pridemo na stran, kjer ima JPlag nadzorni grafični vmesnik za predstavitev svojih rezultatov. Na Sliki 9 vidimo izvirne datoteke, ki se nam izpišejo skrajno levo, na desni strani pa je v odstotkih prikazano število ujemačo se programske kode med dvema datotekama, v odvisnosti od skupne velikosti programske kode v datoteki. Bolj kot se pomikamo v desno, manj enake programske kode posamezne datoteke vsebujejo. Izpis nam prikaže niz ujemanja s povprečno podobnostjo ter niz ujemanja z maksimalno podobnostjo. Povprečna podobnost je običajen postopek, kjer se med seboj preverja celotna programska koda v obeh datotekah. Ujemanja z visoko povprečno podobnostjo kažejo, da programi delujejo na zelo podoben način. Maksimalna podobnost pa je še posebej uporabna, če se datoteke zelo razlikujejo po velikosti. To se največkrat zgodi, ko je vstavljena mrtva koda z namenom, da bi prikrili enakost programske kode v datotekah. Mrtva koda (dead code) je programski ukaz v izvorni kodi programa, ki je izvršen, vendar se ti rezultati nikoli ne uporabljajo. Izvajanje mrtve kode je zgolj zapravljanje časa, saj se njeni rezultati nikoli ne uporabljajo.



Search Results

Matches sorted by average similarity ([What is this?](#)):

Primer1.java	->	Primer6.java (79.7%)	Primer2.java (35.0%)	Primer3.java (18.8%)	Primer4.java (15.5%)
Primer2.java	->	Primer3.java (43.9%)	Primer6.java (31.4%)	Primer4.java (17.2%)	
Primer4.java	->	Primer3.java (20.7%)	Primer5.java (14.2%)	Primer6.java (11.2%)	

Matches sorted by maximum similarity ([What is this?](#)):

Primer1.java	->	Primer6.java (86.0%)	Primer2.java (51.0%)	Primer3.java (23.0%)	Primer4.java (18.0%)
Primer2.java	->	Primer3.java (51.0%)	Primer6.java (51.0%)	Primer4.java (20.0%)	
Primer4.java	->	Primer3.java (21.0%)	Primer6.java (14.0%)	Primer5.java (14.0%)	

Slika 9: JPlag prikaz rezultatov.

S klikom na ime datoteke, ki ni izvorna, se nam prikaže tekst te datoteke in izvorne datoteke (Slika 10). Z modro barvo je označena programska koda, ki pomeni plagiat. Zlasti za večje datoteke je za boljšo preglednost dodana še tabela, ki izpiše, v katerih vrsticah se pojavi plagiatorstvo ter koliko simbolov zajema.

Matches for Primer6.java & Primer4.java

11.2%

[INDEX - HELP](#)

Primer6.java

```

/**
 * @(#)Primer6.java
 *
 *
 * @author
 * @version 1.05 2014/3/24
 */
import java.util.Stack;

public class Primer6 {

    public static void main(String []args) {
        Stack<Integer> sklad = new Stack<Integer>();
        //napolnimo sklad
        sklad.push(0);
        sklad.push(9);
        sklad.push(7);
        sklad.push(2);
        sklad.push(5);

        minNavrh(sklad);
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();

        uredi(sklad);
        System.out.print("Sklad nara?u?o?e: ");
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();
    }
}

```

Primer6.java (9.0%)	Primer4.java (14.0%)	Tokens
Primer6.java(17-26)	Primer4.java(16-52)	8

```

private int maxSize;
private long[] stackArray;
private int top;
public Primer4(int s) {
    maxSize = s;
    stackArray = new long[maxSize];
    top = -1;
}
public void push(long j) {
    stackArray[++top] = j;
}
public long pop() {
    return stackArray[top--];
}
public long peek() {
    return stackArray[top];
}
public boolean isEmpty() {
    return (top == -1);
}
public boolean isFull() {
    return (top == maxSize - 1);
}

public static void main(String[] args) {
    // TODO code application logic here

    int x = 0;
    int s = 0;

    Primer4 theStack = new Primer4(10);
    theStack.push(10);
    theStack.push(20);
    theStack.push(30);
    theStack.push(40);
    theStack.push(50);
    theStack.push(70);
    theStack.push(80);
    theStack.push(90);
    while (!theStack.isEmpty()) {
        long value = theStack.pop();
        System.out.print(value);
        System.out.print(" ");
    }
}

```

Slika 10: JPlag podrobnejši prikaz rezultatov.

Orodje Jplag ničesar ne primerja z datotekami na internetu. Zasnovan je bil tako, da bi v programski kodi našel podobnosti med rešitvami študentov, kar običajno zadošča za odkrivanje plagiatorstva v računalniških programih. Tako deluje tudi danes, le različice programa se stalno dopolnjujejo.

4.1.4 Marble

Orodje Marble [12] je razvil Jurriaan Hage, na Univerzi v Utrechtu na Nizozemskem. Razvijanje je bilo namenjeno predvsem za pomoč predavateljem pri odkrivanju plagiatorstva v programski kodi za programska jezika C# in Java. Kasneje so omogočili odkrivanje plagiatorstva še za nekatere druge programske jezike in tako orodje uspešno uporabljajo še dandanes.

Marble uporablja pristop, ki temelji na strukturi primerjavi (pregleda mape in njihove podmape). Datotečni sistem je v Unixu organiziran hierarhično (drevesno), zato se ob primerjanju datotek preprosto spuščamo navzdol po drevesni strukturi datotečnega sistema in te primerjamo med seboj (Slika 11).

To orodje razcepi predloženo izvorno kodo datotek v ločene datoteke na tak način, da vsaka datoteka vsebuje samo en razred na najvišji ravni. Ko se to izvede, se lahko podrobnosti enostavno spreminjajo ali odstranijo iz teh datotek. Temu postopku pravimo normalizacija. Ta podrobno pregleda datoteke ter izvede leksikalno analizo (razčleni tok znakov na lekseme (rezervirane besede programskega jezika, oznake spremenljivk, ločila itd.)). Za to izvedbo orodje uporablja visokonivojski programski jezik Perl, ki se v splošnem uporablja za programiranje skript. Njegova primerjava temelji na žetonih.

Tukaj se ključne besede npr. Class (razred) in pogosto uporabljeni razredi npr. (String (niz) in System (sistem)) ter imena metod (ToString (predstavi kot besedilo vrednosti)) ohranijo. Komentarji, odvečni presledki, nizi konstant in uvozne deklaracije pa se preprosto odstranijo. Drugi znaki, ki se ohranijo, pa so predstavljeni z njihovimi simboličnimi tipi. Na koncu Marble ustvari dve normirani različici vsake datoteke: prva vsebuje vrstni red polj, metod in notranjih razredov, ki je natanko enako razporejeno kot v originalni datoteki, druga pa vsebuje lastnosti, metode in notranje razrede skupaj razporejene in sortirane. Razvrščanje se izvaja na hevrističen način. Npr. metode so najprej razporejene po številu žetonov, nato po dolžini skupnih žetonov in na koncu še po abecedi. Te datoteke se avtomatsko shranijo v mapo poleg originalnih datotek, ki smo jih določili za pregled.

Primerjava dveh normaliziranih datotek se izvaja z uporabo Unix/Linux ukaza diff. Rezultat se nato izračuna iz razmerja med številom vrstic, na katerih se razlikujejo druga od druge, in skupnega števila vrstic v obeh normaliziranih datotek, ki smo jih primerjali.

```

anej@ubuntu:~/Naloge/Programiranje$ normalizeall.pl
main::doDir() called too early to check prototype at /usr/local/bin/normalizeall.pl line 36.
Only files with at least 30 lines will be done
Doing 2014/APS1
Doing 2014/APS2
Doing 2014/SPO
anej@ubuntu:~/Naloge/Programiranje$ █

```

Slika 11: Marble normalizacija datotek.

Orodje razlikuje med starimi datotekami in datotekami, ki smo jih pravkar dodali za pregled (Slika 12). To omogoča, da starih datotek ne primerjamo ponovno med seboj, ampak zgolj nove datoteke s starimi. Na ta način ne izgubljam časa z odvečnim pregledovanjem.

```

anej@ubuntu:~/Naloge/Programiranje$ detectfraud-shell.pl nf
Run 'suspects.nf' from the current directory to consider all fraud cases in order of descending suspectness.
Starting among the new assignments themselves
* ../Programiranje/2014/APS1/
* ../Programiranje/2014/APS2/
* ../Programiranje/2014/SPO/
Starting the history check
History compare [1/6]
History compare [2/6]
History compare [3/6]
History compare [4/6]
History compare [5/6]
History compare [6/6]
Short summary for reference:
*There were 11 file to file comparisons made
*The following hitvalues are between 0-100 inclusive
*The higher the hitvalue, the less likely the fraud.
*the lowest 0.05 percent of those had hitvalues less than 24
*the lowest 0.1 percent of those had hitvalues less than 25
*the lowest 1 percent of those had hitvalues less than 26
*the lowest 10 percent of those had hitvalues less than 27
Sorting output file. Most likely frauds come first
anej@ubuntu:~/Naloge/Programiranje$ █

```

Slika 12: Marble pregled o vsebovanosti plagiatorstva.

Ko izvedemo ukaz za odkrivanje plagiatorstva, se nam rezultati pristnosti shranijo v datoteko. Nato z urejevalnikom besedil odpremo to datoteko in si pogledamo poročilo. Datoteki, ki sta si najbolj podobni, se nahajata v zapisu na prvem mestu. Izpis prikaže ujemanje datotek v obeh smereh, koliko znakov vsebuje posamezna datoteka, s črkami S in U pa pove, na kakšen način je bila koda razmetana okoli (Slika 13). Črka U pomeni, da je bilo pri kopiranju narejenih zelo malo sprememb, saj se že na oko vidi, da je koda skoraj ista. Črka S pa pomeni, da je bila programska koda zelo spremenjena. Zadnja dva stolpca pa prikazujeta pot in ime datotek, ki smo ju primerjali med seboj.

```

echo 090 269 297 062 S && edit ../Programiranje/2014/APS1/Primer1.java && edit ../Programiranje/2014/SPO/Primer6.java
echo 054 269 243 052 U && edit ../Programiranje/2014/APS1/Primer1.java && edit ../Programiranje/2014/APS2/Primer3.java
echo 054 269 208 049 U && edit ../Programiranje/2014/APS1/Primer1.java && edit ../Programiranje/2014/APS2/Primer4.java
echo 052 243 297 050 S && edit ../Programiranje/2014/APS1/Primer3.java && edit ../Programiranje/2014/SPO/Primer6.java
echo 050 208 297 041 S && edit ../Programiranje/2014/APS2/Primer4.java && edit ../Programiranje/2014/SPO/Primer6.java

```

Slika 13: Marble datotečni izpis o vsebovanosti plagiatorstva.

Kako se programi razlikujejo, lahko pogledamo tudi z urejevalnikom besedila Vim. Ta je zelo uporaben tudi za programerje, saj lahko z ukazom »vimdiff« primerjamo dve ali največ tri datoteke med seboj. Pri izpisu pridobi vsaka datoteka svoje okno. Razlike med datotekami se poudarijo z več barvnimi odtenki. To je lep grafični način za ugotovitev razlik in enakosti, ki jih datoteke vsebujejo (Slika 14).

```

1.06 2014/3/24
*/
import java.util.Stack;

-----
public class Primer1 {
    public static void main(String []args) {
        Stack<Integer> sklad = new Stack<Integer>();
        //napolnimo sklad
        sklad.push(4);
        sklad.push(2);
        sklad.push(1);
        sklad.push(3);
        sklad.push(5);
        sklad.push(9);
        sklad.push(3);
        sklad.push(5);
        sklad.push(7);

        /* izprazni sklad */
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();

        uredi(sklad);

        System.out.print("Sklad nara<9a>eujoée: ");
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();
    }
}
2014/APS1/Primer1.java 6,1

1.05 2014/3/24
*/
import java.util.Stack;

-----
public class Primer6 {
    public static void main(String []args) {
        Stack<Integer> sklad = new Stack<Integer>();
        //napolnimo sklad
        sklad.push(6);

        sklad.push(9);

        sklad.push(7);
        sklad.push(2);
        sklad.push(5);

        minNaVrh(sklad);
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();

        uredi(sklad);

        System.out.print("Sklad nara<9a>eujoée: ");
        while(!sklad.empty())
            System.out.print(sklad.pop()+" ");
        System.out.println();
    }
}
2014/SPO/Primer6.java [R0] 6,1 8%

```

Slika 14: Marble podrobnejši prikaz podobnosti.

Marble je brezplačno orodje, vendar ga na internetu za prenos ne zasledimo. Če ga želimo pridobiti, je potrebno dogovorjanje z njegovim avtorjem. Ni odprtokodni sistem.

4.1.5 Povzetek zgoraj opisanih orodij za detekcijo v programski kodi

Pri orodjih za odkrivanje podobnosti v programski kodi ne moremo reči, katera so boljša in katera slabša. Vsako ima svoje prednosti in slabosti.

Najprej jih lahko preverjamo po številu podprtih programskih jezikov, ki nam jih omogoča posamezno orodje. Nekatera orodja nam omogočajo razširljivost, kar pomeni, da lahko sami dodamo poljuben programski jezik v orodje ter preverjamo podobnost programske kode v na novo dodanem programskem jeziku.

Pri nekaterih orodjih je omogočeno tudi to, da lahko predhodno določimo, katera programska koda je vnaprej podana. Tu gre predvsem za kodo, ki jo profesorji podajo študentom in služi kot osnova za delovanje programa. Tako kodo lahko nekatera orodja spregledajo, kar se doseže v okviru nastavitvev. Na ta način se izognemo lažnim pozitivnim rezultatom enakosti, ki se štejejo za plagiatorstvo.

Veliko orodij za odkrivanje plagiatorstva ima možnost izključitve iz pregleda zelo majhne datoteke, kot je npr. javansko zrno (JavaBean), saj vsebujejo le svoje attribute in get ter set metode. Če takih datotek ne bi izvzeli iz preverjanja, bi program vrnil visok odstotek podobnosti, saj bi menili, da je bila programska koda v teh datotekah prepisana. Tako pa nam prikaže zgolj dejansko stanje vsebovanosti enake programske kode.

Za pregled zelenih datotek lahko nekaterim orodjem izberemo več datotek hkrati v posamezni mapi. Nekatera pa imajo možnost izbrati celotno mapo z vsemi podmapami v izbranem imeniku datotečnega sistema ter tako preverjamo podobnosti med datotekami v mapah. Ta možnost je zelo uporabna pri profesorjih, ki morajo primerjati večje število map, saj lahko na ta način primerjajo letošnje oddaje programske kode z lanskimi, oddaje posamezne skupine z ostalimi ter tako najlažje ugotovijo, če je šlo za plagiatorstvo.

Da skrajšamo čas iskanja plagiatorstva, imajo nekatera orodja sposobnost vpogleda v zgodovino primerjav. To možnost imajo orodja, kjer lahko podamo za iskanje zelena področja v imeniku (mape in podmape). V primeru, da smo določeno mapo že pregledali in so rezultati poročila shranjeni v zgodovini primerjanja, te mape ne pregledamo ponovno, temveč vzamemo stare rezultate preverjanja in te preverjamo z novo izbrano mapo (vključno s podmapami).

Nekatera orodja se izvajajo lokalno, spet druga so na voljo kot spletne storitve. Tista, ki se izvajajo kot spletne storitve, zahtevajo pošiljanje informacij po omrežju spletnemu strežniku. Lokalno izvajanje zahteva namestitev orodja ter zaganjanje na lokalnem računalniku.

Poznamo tudi odprtokodna orodja, ki imajo možnost razširitve ali izboljšanja programa. Tako se program lahko priredi specifični situaciji, ko moramo preverjati plagiatorstvo.

Za primerjanje zgoraj opisanih funkcionalnosti v orodjih prilagam Tabela 1:

Orodje	MOSS	SIM	Jplag	Marble
Število podprtih programskih jezikov	26	7	6	2
Razširljivost	ne	da	ne	ne
Izključuje vnaprej podano programsko kodo	da	ne	da	ne
Izključuje majhne datoteke	da	ne	da	da
Pregled datotek ali map	datoteke	mape	datoteke	mape
Lokalno ali spletno orodje	spletno	lokalno	spletno	lokalno
Zgodovina preverjanja	ne	da	ne	da
Operacijski sistem	Windows ali Unix/Linux	Windows ali Unix/Linux	Windows ali Unix/Linux	Unix/Linux
Odprtokodni sistem	ne	da	ne	ne
Prikaz rezultatov ¹	odlično	zadovoljivo	odlično	dobro
Uporabnost ¹	dobro	dobro	odlično	zadovoljivo

Tabela 1: Primerjava orodij za odkrivanje plagiatorstva v programski kodi. [23]

V Tabeli 1 sem izrazil svoje mnenje tudi glede uporabnosti samega orodja ter način, kako je poročilo o plagiatorstvu prikazano. Pri uporabnosti sem upošteval, kako zahtevno je orodje za uporabnika, koliko časa potrebuje, da ga usposobi za delo, ali uporablja grafični vmesnik ali ukazno vrstico ...

Pri prikazu rezultatov je pomembno, da prikaže povzetek o vseh datotekah, ki smo jih primerjali. Veliko pripomore tudi prikaz podrobnosti dveh programskih kod med seboj, kjer so deli enake programske kode dodatno označeni. Tako lahko uporabnik hitro ugotovi, ali gre res za plagiatorstvo ali ne. Pri rezultatih se pričakuje tudi podrobnejši opis, da jih lahko vsak uporabnik razume. Če orodje uporablja za prikaz tudi grafikone, kot je npr. histogram, je to še dodatna prednost.

¹ Za orodja, ki so se po testiranju izkazala za najboljša sem uporabil izraz "odlično", za povprečna orodja sem uporabil izraz "dobro", za slabša orodja pa sem uporabil izraz "zadovoljivo".

4.1.6 Priporočilo

Po vseh testiranjih, ki sem jih izvedel, bi za programsko kodo priporočil uporabo orodja Jplag, saj je uporabniški vmesnik zelo preprost in pregleden.

Poganjamo ga lahko na Windows ali Unix/Linux operacijskih sistemih, saj za uporabo njihovih spletnih servisov potrebujemo zgolj namestitvev aplikacije klienta. Čeprav se pregled podobnosti datotek izvaja na strežniku, pridemo do rezultatov razmeroma hitro.

Rezultati, ki jih orodje prikaže, so zelo zanesljivi ter prikazani so razumljivo in razločno. K temu veliko pripomore tudi palični grafikon, saj lahko že iz grafikona vidimo, kje je bilo odkritih največ enakosti v programski kodi. Poleg tega pa lahko datoteke, ki imajo večji odstotek ujemanja, primerjamo med seboj. To primerjanje je uporabniku zelo lepo prikazano, tako da lahko hitro ugotovi, ali je šlo za plagiatorstvo ali pa ne. Za lažje razumevanje rezultatov je v menijih omogočena tudi pomoč v obliki podrobnejših opisov.

Ena od prednosti je tudi ta, da lahko do starih preverjanj dostopamo na zelo preprost način, saj se le-ta hranijo v zapisu za spletne strani na našem lokalnem disku v mapi "JPlag_Results", ki jo orodje samo ustvari.

Čeprav ima podporo za manj programskih jezikov kot npr. sistem MOSS, sem prepričan, da se bo to sčasoma spremenilo ter bodo dodani še ostali programski jeziki.

5 Odkrivanje plagiatorstva v prostem tekstu

Pred prihodom računalniško podprtih orodij in storitev se je plagiatorstvo odkrivalo ročno s primerjanjem virov. To je bilo izjemno zamudno početje, saj je bilo potrebno obravnavano delo primerjati z drugimi deli in iskati podobnosti ali enakosti med njima. V primeru, da je bilo delo zelo obsežno, so bile tudi možnosti za odkritje plagiatorstva izredno majhne.

Z uporabo računalnika si lahko pomagamo tudi tako, da več sumljivih besed ali stavkov ročno vpišemo ali kopiramo v spletni iskalnik in pregledamo rezultate, ki nam jih vrne iskalnik. Če se med rezultati iskanja besede ali stavki ujemajo, lahko po vsej verjetnosti sklepamo, da gre za plagiatorstvo. Da to potrdimo, se moramo še bolj poglobiti v ta spletni vir in preučiti datum nastanka, kdo je avtor ter šele nato lahko potrdimo, da je delo plagiat.

Na srečo imamo dandanes na razpolago tudi veliko število storitev in programov, ki samodejno preverjajo plagiatorstvo v prostem tekstu in nam tako prihranijo veliko dragocenega časa.

5.1 Programi in storitve za detekcijo plagiatov v prostem tekstu

Večina programov ali storitev deluje tako, da pregledajo naš dokument in ga primerjajo z dokumenti, ki so objavljeni na internetu in v spletnih arhivih. Nekateri pa imajo tudi možnost, da pregledajo naš dokument in ga primerjajo z dokumenti, ki so shranjeni v mapah na lokalnem računalniku ali s poljubno izbrano datoteko. Tako izbiro mi je v diplomskem delu omogočal le program Plagiarism Detector. Če se ugotovi podobna kombinacija besed, kot so zapisane v našem dokumentu, se odsek nazorno prikaže ter obarva. Oseba, ki izvaja pregled našega dokumenta, mora nato pregledati označene odseke, da ugotovi, če gre resnično za plagiat. Nekateri programi nam rezultat prikažejo tudi v procentih ter tako še bolj nazorno prikažejo vsebovanost enakega besedila v prostem tekstu. Koliko delov besedila se bo dejansko osvetlilo, je odvisno predvsem od natančnosti iskanja našega programa. Vsak program ima določeno svoje področje iskanja, zato pri njihovem primerjanju pridemo do različnih rezultatov.

Eden najpriljubljenejših programov je Viper, saj ga lahko brezplačno pridobimo na spletu. Program za odkrivanje plagiatorstva pregleda več kot 10 milijard spletnih virov. Ta program se trži predvsem za posameznike, ki bi radi preverili, če je določeno literarno delo že bilo objavljeno na spletnem medmrežju oziroma za preverjanje avtorstva besedila.

Naslednji priljubljeni program je Turnitin, ki ga uporabljajo tudi izobraževalne ustanove. Ponuja možnost nakupa licence, namenjen pa je inštruktorjem in profesorjem. Pred začetkom pisanja diplomskega dela sem si zelo prizadeval, da bi pridobil licenco za uporabo tega programa, vendar mi je kmalu postalo jasno, da to ne bo mogoče, saj Fakulteta za računalništvo in informatiko nima zakupljene licence, za posamezne študente pa program žal ni na voljo. Ta program sicer uporablja več fakultet na Univerzi v Ljubljani (Fakulteta za upravo, Ekonomska fakulteta itd.) .

5.1.1 Plagiarism check.org

Je zelo zanesljiva spletna storitev [31] za odkrivanje plagiatorstva v prostem tekstu. Med njenimi prednostmi bi izpostavil možnost preverjanja neomejenega števila dokumentov, poročilo o plagiatorstvu pa se ustvari izredno hitro tudi v primeru, ko preverjamo več dokumentov naenkrat.

Za uporabo te storitve je obvezna registracija, saj z njo pridobimo svoj uporabniški račun in osebni prostor na strežniku, kjer se hranijo datoteke, ki jih želimo pregledati. Datoteke lahko naložimo v kateri koli obliki (.txt, .doc, .docx, itd.). Lahko pa v polje vpišemo zeleno besedilo in storitev nam to besedilo samodejno shrani na strežnik v tekstovno datoteko (.txt). Ko se pregled datoteke zaključi, se poročilo avtomatično posreduje kot odgovor na elektronsko pošto. V poročilu se nahaja informacija o tem, katera datoteka je bila pregledana in kolikšen procent enakega besedila je bil odkrit na internetu in v spletnih arhivih (Slika 15).

Tako se baza pregledanih datotek stalno povečuje ter nam obenem daje natančen in zanesljiv prikaz poročil o vsebovanosti enakega besedila, kot ga je bilo mogoče zaslediti tudi na svetovnem spletu. Možnost imamo tudi, da si poročila prenesemo na svoj računalnik v obliki .pdf datotek. Storitev uporablja za primerjavo veliko število spletnih virov (pridobi jih z pomočjo spletnih iskalnikov, ki avtomatsko preverijo zaporedje besed ali stavkov, če so se te že pojavile kje na medmrežju). Algoritmi za iskanje plagiatorstva se stalno posodablajo. Lahko bi rekli, da ta storitev ponuja izjemno veliko, glede na to, da je brezplačna.

PLAGIARISM CHECK.org Home Your documents Tour Usage Feedback Settings

Hi, andip. Log Out

Your Documents » **Za_testiranje.txt** Plagiarism approximately **7%** in **6** Sources

Za testiranje plagiatstva, sem se odločil, da opišem zgradbo in delovanje računalnika. Nekaj stavkov bom napisal z svojimi besedami, ostalo vključeno gradivo, pa bom pobral z interneta. To besedilo bom kasneje preverjal z programi za odkrivanje plagiatstva in si njihove rezultate zabeležil, ter jih primerjal med seboj. Računalnik je splošno uporabna naprava, ki se jo lahko programira za izvajanje aritmetičnih in logičnih operacij. Ker se zaporedja izvajanja operacij zlahka spremenijo, lahko računalnik rešuje različne vrste problemov.

Leta 1970 so naredili Imac PDS-1, prvi računalnik, v katerem bi dandanes takoj prepoznali osebni računalnik, saj je imel svoj monitor (z vektorskim, ne rastrskim zaslonom, kakršne imamo dandanes) in tipkovnico, kar za računalnike tistega časa še zdaleč ni bilo kar samo po sebi razumljivo. Sodiši je v razred grafčnih miniračunalnikov in je bil še veliko predrag, da bi mu lahko rekli osebni računalnik. Računalnik je naprava za shranjevanje in obdelavo podatkov in ena od naprav za prenos informacij. Podatek predstavlja dejstva, pojme na formaliziran način, s simboli.

Informacija so ovrednoteni podatki v specifični situaciji (G.C.Everest, 1986) Podatki niso informacije, podatki ne vsebujejo informacije, podatki posredujejo informacije prejemniku, katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta, na katerega se nanašajo, če je količina podatkov tako velika, da se jih v času, ki je na voljo za ukrepanje na njihovi osnovi, ne da interpretirati, se lahko zgodi, da s podatki ni posredovana nobena informacija. (S. Lasgoffers, 1980) Računalniki ali tudi računalniški sistemi so stroji/avtomati z obdelavo informacij sestavljeni iz strojne opreme, programske opreme in komunikacijske opreme. Za procesiranje uporabijo CPU (centralno procesno enoto). Imeti mora vsaj eno takšno enoto.

Centralna procesna enota iz centralnega polnilnika jemlje ukaze in jih izvršuje. Običajno jo delimo na: krmilno – kontrolno enoto, aritmetično – logično enoto in registre. Za povezave med njimi skrbijo notranje podatkovne poti, za povezavo procesorja s polnilnikom in vhodna / izhodni sistemom pa zunanje podatkovne poti. Ali veste, da je vaš mobilni telefon pravzaprav računalnik, ki izvaja programe za digitalno procesiranje signalov? Sedaj tudi to znate. Obseg velikosti in zmogljivosti računalnikov.

Na eni strani so superračunalniki, torej zelo veliki računalniki z na tisoče povezanimi mikroprocesorji, ki izvajajo izredno zapletene izračune. Na drugem koncu pa so zelo majhni računalniki, ki so vgrajeni v avtomobile, televizije, stereo sisteme, računalna in aparate. V teh računalnikih je mogoče izvesti omejeno število opravil. Takšen model je v 40-tih letih prejšnjega stoletja zasnoval John Von Neumann, ameriški kemik in dr. matematike, madžarskega rodu. Prvi je predlagal zapis podatkov v dvojiškem zapisu. Računalniki, ki so jih takrat poznali so dokaj hitro izvajali operacije, problem se je pa pojavil, ker je bilo potrebno vsakik znova vpisovati navodila za njihovo delo.

John von Neumann je prvi uvedel pojem shranjenega programa, kar velja za prelomnico razvoje tehnologije računskih strojev, ki so imeli v pomnilni enoti shranjene le podatke, potrebne za obdelavo, in računalnikov, ki imajo poleg njih shranjena tudi navodila (programi) za njihovo obdelavo. Vhodno-izhodne enote omogočajo izmenjavo informacij med računalnikom in njegovim okoljem. Med vhodne naprave štejemo računalniško miško, tipkovnico, grafično tablico, ...

Izhodne naprave pa so zaslon, tiskalnik, zvočniki, ... Programska oprema je skupek programov, ki jih uporabljamo pri delu z računalnikom. Verjetno vsi poznate občutek, ko računalnik nenadoma ne deluje več dovolj hitro in nam povzroča sve lasa, medtem ko s poslušanjem "graskanja" po disku čakamo, da zvede zahtevano operacijo, klavirno je to znak, da je napočil čas za formatiranje diska, s čimer ga očistimo navlake ali za nakup novega računalnika. Prenosni računalnik ali notesnik je manjši in lažji prenosljiv osebni računalnik. Za razliko od namiznega osebnega računalnika ima vgrajeno baterijo z možnostjo polnjenja. Baterija omogoča uporabo brez električnega napajanja iz omrežja.

Sources found:

Source	Percentage
all sources	7%
www.monitor.si	6%
www2.arnes.si	5%
computer.howstuffworks.com	3%
urbizefran.blogspot.com	3%
prenosali.eu	1%
projekti.gimvic.org	1%

Slika 15: Plagiarism check.org prikaz poročila o ujemanju besedila na internetu.

5.1.2 Plagiarism checker

Dobimo ga med orodji Small SEO Tools, ki nam ponujajo več različnih brezplačnih spletnih orodij, kot so štetje besed, preverjanje črkovanja, preurejanje člankov, pa tudi več rešitev za spletne storitve. Za potrebe diplomskega dela sem uporabljal predvsem storitev Plagiarism checker [37], ki s pomočjo iskalnika Google odkriva plagiatstvo v prostem tekstu.

Tekst, ki ga hočemo preveriti, se vnese v polje za vnos. Vpisani tekst se nato razdeli na majhne fraze, nakar se besedilo preveri s pomočjo svetovnega spleta (uporablja vire, ki jih pridobi z pomočjo spletnega iskalnika Google, ki avtomatsko preverja zaporedje fraz, če so se te že kje pojavile na medmrežju). V primeru, da se pojavi enaka fraza, bo storitev to prepoznala. Taki stavki ali zaporedja besed se prikažejo pod vnesenim tekstom in se obarvajo rdeče ter vsebujejo URL povezavo, kjer se nahaja dobljena vsebina. Vsebina na spletu je lahko avtorsko delo, lahko pa je že plagiatorsko delo, kar moramo sami še dodatno preveriti. Ko se preverjanje zaključi, nam storitev prikaže odstotek našega unikatnega besedila, ostalo je plagiat.

Storitev ima omejitve in sicer dnevno lahko preverimo 3.000 besed. Če to število presežemo, nam storitev preneha delovati za 24 ur, po preteku tega časovnega intervala pa spet deluje normalno. Poleg tega v storitev lahko vnesemo le besedilo in ne celotnih datotek, kar prikazuje Slika 16.

Software is the name given to programs, which are running on the computer. First, we have operating system, such as Apple OS, Linux, Windows XP, Windows Vista, Windows 7 or Windows 8. Then we could install other software, like the games or the other programs for work.

Tablets have become one of the fastest selling computing devices in history. Their compact size and low costs make them appealing to many people compared to laptops. Their media capabilities in particular are helping them move more people away from traditional TV. So if you happen to be looking at picking up a tablet, check out my guide to help determine which may be the best one to fit your needs.

Maximum 1500 words limit per search. Total Words: 1000

431x99

Can't read the image? click [here](#) to refresh

Check for Plagiarism

Results: 46% Unique Content

or Windows 8. Then we could install other software, like	Good
other software, like the games or the other programs for	Good
devices in history. Their compact size and low costs make them	Existing (Mar 6, 2014)
compared to laptops. Their media capabilities in particular are	Existing (Mar 6, 2014)
traditional TV. So if you happen to be looking at picking	Existing (Mar 6, 2014)
picking up a tablet, check out my guide to help determine which	Existing (Mar 6, 2014)

A part of your content is matching some text from the Web. Please make sure that this text is correctly referenced.

Slika 16: Plagiarism checker prikaz poročila o ujemanju besedila na internetu.

5.1.3 PlagScan

To je še ena spletna storitev, ki preverja pristnost dokumentov (MS Word, .pdf, .txt, itd.). Druga možnost je, da se besedilo vpiše neposredno v prostor za dodajanje teksta ter se na strežniku samodejno shrani kot datoteka za preverjanje pristnosti. Storitev PlagScan [29] uporablja zelo napreden dvostopenjski algoritem, ki temelji na najnovejših raziskavah na področju računalniške lingvistike.

Storitev je zelo priljubljena, saj jo uporablja več kot tisoč organizacij na vseh celinah. Za pregled plagiatorstva vključi v preverjanje lokalno bazo dokumentov ter dokumente hranjene na medmrežju, ki jih skupaj vsebujeta več kot milijardo. Storitev PlagScan ima svojo lastno bazo podatkov, ki vključuje dokumente kot so knjige, članki in ostala gradiva. Poleg lastne baze preverja pridobljene rezultate tudi s pomočjo spletnega iskalnika Yahoo, ki avtomatsko preveri zaporedje besed ali stavkov, če so se te že pojavile kje na medmrežju (vključno s spletnimi dokumenti v .pdf oblikah, kot so knjige, članki, revije, časopisi, itd.). Ta baza podatkov se dnevno dopolnjuje. Na neodvisnih testiranjih leta 2013 je spletna storitev PlagScan osvojila prvo mesto med mednarodno konkurenco z vidika uporabnosti pri odkrivanju plagiatorstva [34].

Poročila o plagiatu izpolnijo skoraj vsako zahtevo uporabnika. Že v glavnem meniju nas poročilo seznanja z ujemanjem rezultatov po primerjanju z lokalno bazo podatkov ter s spletnim ujemanjem. Ti rezultati so prikazani tudi z lestvico, ki prikazuje raven plagiatstva in jo storitev avtomatsko prikaže ter določi po svojih kriterijih (Slika 17). Če se povezava za pregled poročila obarva zeleno, pomeni, da je zagotovo naše lastno delo. Če se obarva rumeno, pa je že priporočen natančnejši pregled besedila, saj je storitev odkrila podobno besedilo, kar bi lahko pripeljalo do potrditve plagiatstva. Če se obarva rdeče, pa gre skoraj zagotovo za plagiat, ki pa ga je potrebno naknadno še potrditi.

The screenshot shows the PlagScan web interface. At the top, there is a navigation bar with 'PlagScan', 'Documents', 'Settings', and 'Logout'. On the left, a sidebar contains user information and navigation options. The main area displays a table of documents with columns for document name, word count, similarity percentage, and date. A legend at the bottom left indicates plagiarism levels: 0-1% (green), 1-5% (yellow), and 5-100% (red).

Add document(s)		2 documents	
Select all	Selected: 0	Check	Delete
<input type="checkbox"/>	Za testiranje plagiatstva2.doc - Za testiranje plagiat	1000 words	44.9% Report-> 2014-03-17 16:02
<input type="checkbox"/>	Unnamed - JavaScript Object Notation (JSON) je pre	136 words	Start check (2) 2014-01-15 12:19

Hint: Checks continue on our servers even if you log out of PlagScan.

Plagiarism level

- 0-1%
- 1-5%
- 5-100%

Slika 17: PlagScan prikaz ravni plagiatstva.

Da se še sami prepričamo v te rezultate, imamo na voljo povezave, kjer se lahko seznanimo s seznamom zadetkov (ujemanje besedila na spletu ter lokalnih bazah), interaktivnim poročilom (prikaz našega ter kopiranega besedila) ter imamo možnost, da si poročilo prenesemo kot standardno datoteko za izmenjavo elektronskih dokumentov (.pdf) ali Microsoft Word dokument (vključno s pripombami o plagiatstvu).

Gre za časovno učinkovito storitev in enostavno za uporabo, saj lahko naenkrat naložimo celotno mapo (.zip) arhivov, ki vsebujejo datoteke z besedilom. Arhivska mapa se nam postavi v glavni meni, njene tekstovne datoteke pa se nam izpišejo in prikažejo v vrsti za pregled plagiatstva. Čas za analizo besedila je odvisen od velikosti besedila. Ponavadi to traja nekaj minut, večji dokumenti pa so zaključeni najkasneje naslednji dan.

Bogato strukturirana poročila o plagiatstvu omogočajo hitro spremljanje. Poročilo o plagiatstvu je po zaključku pregleda podrobneje prikazano kot HTML dokument (Slika 18). Tukaj so prikazani vsi rezultati ujemanja, ki nam tako hitro omogočajo ugotoviti odlomke, iz katerih spletnih naslovov je bilo mogoče besedilo prepisano. Za obsežno primerjavo kliknemo na določen vir in ta nas postavi na spletno mesto, kjer je vsebina zapisana ter tako lahko pričnemo z neposredno primerjavo.

Filename: Za testiranje plagiatstva2.doc Date: 2014-03-17 16:02
Results of plagiarism analysis from 2014-03-17 16:09 PlagScan

81 matches from 39 sources, of which 39 are online sources. Plagiarism level: 44.9%

[Print](#) | [Download as PDF](#) | [View within text](#) | [Word docx document with annotations](#)

Showing best match per source - click on match to display all matches or on URL to show source.

5 matches from <http://www.racunalniske-novice.com/forum/topic/16953-opis-racunalnika/> [Highlight matches in source](#)

- ... Računalniki ali tudi računalniški sistemi so stroji avtomati z obdelavo informacij sestavljeni iz programske opreme ... programske opreme in komunikacijske opreme ...
- Za povezave med njimi skrbijo ... poti za povezavo procesorja s pomnilnikom in vhodna izhodna sistemom pa zunanje ...
- ... procesna enota iz centralnega pomnilnika jemlje ukaze in jih izvršuje
- kontrolno enoto aritmetično ... logično enoto in registre
- Običajno jo delimo na

4 matches from http://www.djaski.net/get/rif_plo_racunalniki_informatika_zgodovina_racunalnikov_01.doc [Highlight matches in source](#)

- ... niso informacije podatki ne vsebujejo informacije podatki posredujejo informacijo prejemniku katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta na katerega se nanašajo če je količina podatkov tako velika da se jih v času ki je na voljo za ukrepanje na njihovi osnovi ne da interpretirati se lahko zgodi da s podatki ni potredovana nobena informacija ...
- Računalnik je naprava za shranjevanje in obdelavo podatkov in ena od naprav za prenos informacij
- Podatek predstavlja dejstva pojme na formaliziran način s simboli
- Informacija so ovrednoteni podatki v specifični situaciji G ...

4 matches from http://www.djaski.net/get/rif_plo_racunalniki_informatika_zgodovina_racunalnikov_01.pdf [Highlight matches in source](#)

- ... niso informacije podatki ne vsebujejo informacije podatki posredujejo informacijo prejemniku katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta na katerega se nanašajo če je količina podatkov tako velika da se jih v času ki je na voljo za ukrepanje na njihovi osnovi ne da interpretirati se lahko zgodi da s podatki ni potredovana nobena informacija ...
- Računalnik je naprava za shranjevanje in obdelavo podatkov in ena od naprav za prenos informacij
- Podatek predstavlja dejstva pojme na formaliziran način s simboli
- Informacija so ovrednoteni podatki v specifični situaciji G ...

4 matches from <http://windows.microsoft.com/si-si/windows/introduction-to-computers> [Highlight matches in source](#)

- Na eni strani so superračunalniki torej zelo veliki računalniki z na tisoče povezanimi mikroprocesorji ki izvajajo izredno zapletene izračune
- Na drugem koncu pa so zelo majhni računalniki ki so vgrajeni v avtomobile televizije stereo sisteme računalna in aparate
- V teh računalnikih je mogoče izvajati omejeno število opravil ...
- Obseg velikosti in zmožljivosti računalnikov

4 matches from <http://www.antessays.com/free-essays/414785.html> [Highlight matches in source](#)

- A microprocessor is a small electronic device that can carry out complex calculations in the blink of an eye ...

4 matches from <http://www.docstoc.com/docs/124815708/Computer---How-Stuff-Works> [Highlight matches in source](#)

- A microprocessor is a small electronic device that can carry out complex calculations in the blink of an eye ...

Slika 18: PlagScan podrobnejši prikaz ujemanja našega besedila z besedilom na svetovnem spletu.

Lahko pa si ogledamo interaktivno poročilo, ki nam s pomočjo brskalnika omogoča pregled odkritih ujemanj neposredno v besedilu. V več barvah nam izpiše dobesedna ujemanja, kar se verjetno nanaša na vir, vendar je bilo spremenjeno ter citiranja (Slika 19). Poleg tega nam prikaže tudi padajočo lestvico ujemanja sobesedila s spletnimi viri podano v procentih ter tako najlažje ugotovimo, na katerem URL naslovu je bilo največ ujemanj.

Filename: Za testiranje plagiatorstva2.doc Date: 2014-03-17 16:02
Results of plagiarism analysis from 2014-03-17 16:09 PlagScan

81 matches from 39 Sources, of which 39 are online sources. PlagLevel: 47.2%

Print | Download as PDF | Show Comparison | View as list | Docx document with annotations

- [0] (5 Matches, 6.3%) www.racunalniske-novice.com/forum/topic/16953-opis-racunalnika/ [Highlight matches in source](#)
- [1] (4 Matches, 9.2%) www.djaski.net/get/rnf_plo_racunalniki_informatika_zgodovina_racunalnikov_01.doc [Highlight matches in source](#)
- [2] (4 Matches, 9.2%) www.djaski.net/get/rnf_plo_racunalniki_informatika_zgodovina_racunalnikov_01.pdf [Highlight matches in source](#)
- [3] (4 Matches, 5.6%) windows.microsoft.com/sl-si/windows/introduction-to-computers [Highlight matches in source](#)
- [4] (4 Matches, 5.7%) www.docstoc.com/docs/124815708/Computer---How-Stuff-Works [Highlight matches in source](#)

29 Matches Citation Remove marking Save changes

Analyzed document

Za testiranje plagiatorstva, sem se odločil, da opišem zgradbo in delovanje računalnika. Nekaj stavkov bom napisal z svojimi besedami, ostalo vključeno gradivo, pa bom pobral z interneta. To besedilo bom kasneje preverjal z programi za odkrivanje plagiatorstva in si njihove rezultate zabeležil, ter jih primerjal med seboj.

Računalnik

je splošno uporabna naprava, ki se jo lahko programira za izvajanje aritmetičnih in logičnih operacij. Ker se zaporedja izvajanja operacij zlahka spremeni, lahko računalnik rešuje različne vrste problemov.

Leta

1970 so naredili imlaci PDS-1, prvi računalnik, v katerem bi dandanes takoj prepoznali osebni računalnik, saj je imel svoj monitor (z vektorskim, ne rastrskim zaslonom, kakršne imamo dandanes) in tipkovnico, kar za računalnike tistega časa še zdaleč ni bilo kar samo po sebi razumljivo. Soditi je v razred grafičnih miniračunalnikov in je bil še veliko predrag, da bi mu lahko rekli osebni računalnik.

Računalnik

je naprava za shranjevanje in obdelavo podatkov in ena od naprav za prenos informacij. Podatek predstavlja dejstva, pojme na formaliziran način, s simboli. Informacija so ovrednoteni podatki v specifični situaciji (G.C.Everest, 1986)

Podatki

Legend text highlighting
Aa exact match
Aa possibly altered text
Aa marked as quotation

Slika 19: Prikaz našega dokumenta vključno z obarvanimi stavki, ki se štejejo za plagiatorstvo.

Storitev tudi omogoča, da si rezultate poročila shranimo v obliki MS Word datoteke, kjer bo zapis vsebine izvirnega dokumenta vključeval rezultate ujemanja ter jih ustrezno označil in zabeležil. Ta vrsta poročila ponuja elegantno in udobno rešitev, saj skoraj nikoli ne spreminja oblike dokumenta. Tako nastane enak dokument z dodanimi komentarji. Poleg tega si s tem pogledom prihranimo ogromno časa, saj omogoča preprost način za odkrivanje in določitev plagiatorstva.

Storitev PlagScan ni brezplačna. Ob prvi registraciji prejmemo zadostno število kreditov, ki nam omogočajo pregled plagiatorstva za 2.000 besed. Za večje število besed pa so potrebna doplačila.

5.1.4 Viper

Viper [32] je med naj priljubljenejšimi programi za odkrivanje plagiatorstva v prostem tekstu, saj je zelo natančen, brezplačen in precej enostaven za uporabo. Prikaže nam označeno primerjavo katerega koli ujemanja našega besedila z najdenim virom na svetovnem omrežju. Programa ne uporabljajo zgolj učitelji, ampak je v pomoč tudi študentom, da ugotovijo izvor prepisanega gradiva ter ga lahko nato pravilno citirajo.

Začetki programa Viper segajo v leto 2007, ko ga je prvič imenovala zasebna podporna izobraževalna služba All Answers Limited iz Velike Britanije, ki zagotavlja akademske podporne storitve za študente. Prve različice programa so napisali ukrajinski razvijalci programske opreme z namenom, da bi ekipa za nadzor kakovosti v raziskovalnem delu prepoznala plagiatorstvo v literaturi, ki jo je podjetje prejelo.

Programska oprema je bila poleg uporabe za zaposlene v podjetju ponujena tudi drugim raziskovalcem izven podjetja, da se jim je omogočalo preverjanje pristnosti lastnega dela pred oddajo. Sčasoma je podjetje spoznalo, da bi lahko tudi ostali želeli preverjati originalnost raziskovalnega dela, zato so jim v ta namen ponudili programsko opremo.

Ko je popularnost programa narasla, so se v podjetju odločili, da povsem brezplačno ponudijo produkt še ostalim uporabnikom (študentom, učiteljem in ostalim uporabnikom, ki želijo uporabljati ta program).

Ker je uporabljalo program že več tisoč uporabnikov, funkcionalnosti pa se niso kaj prida izpopolnile, se je podjetje v juniju 2009 odločilo, da ustanovijo notranjega razvijalca programske opreme. Sledili sta dve izdaji, ki sta imeli stabilnejši in zanesljivejši sistem ter dodatne funkcije in izboljšano funkcionalnost.

Oktober leta 2010 so program Viper (skupaj s svojimi razvijalci programske opreme) priključili podjetju All Answers Limited. Z ekipo šestih ljudi so skrbeli za programsko opremo in spletne strani. Storitve ni bila več brezplačna, ampak si za uporabo moral plačati nizkocenovno pristojbino za želeno število mesecev. Te različice so bile na voljo do aprila 2011.

Kasneje je program Viper postal brezplačna različica programa Turnitin, ki je dandanes temelj za preverjanje plagiatorstva v šolstvu, saj ga uporablja večina univerz in visokih šol po vsem svetu.

Njegovi uporabniki so učenci, učitelji (predavatelji), lastniki spletnih strani in vladni uradniki. Pri preverjanju plagiatorstva preveri okrog 10 milijard spletnih virov, kot so spletne revije in knjige ter nato rezultate nazorno prikaže. Pri vsakem odkritju enakega dela besedila se izpiše URL povezava do strani, ki vsebuje enako besedilo (Slika 20). Zraven pa je še pripisano, koliko besed se na dobljeni spletni strani ujema z našim besedilom.

THE ANTI-PLAGIARISM SCANNER THAT'S ACCURATE, EASY AND FREE!

Viper (4.1.90.1039)

File Actions Reports Options Help

Viper THE ANTI-PLAGIARISM SCANNER

Files to Scan

Filename	Category	Word Count	Unique Words	Queries	Plagiarism (%)	Scan Time	Status	Progress	Print	Delete
Za testira	Comput	996	606	64	31	00:02:11	Finished	100%		

Scanning in progress: 1 / 1

Found Documents

Location	Title	Words Matched	Match (%)	Unique Words Matched	Unique Match (%)
http://compreviews.about.com/	PC Hardware and Reviews - Computer Reviews and Information on Desktops, Laptops and Components Computing PC Hardware Reviews Search PC Hardware / Reviews Buyer's Guides PC Reviews Build A PC Share Must Reads What PC Should I Buy Tablet Buyer's Guide Laptop PC Buyer's Guide Desktop PC Buyer's Guide Best Chromebooks Guide since 2002 Mark Kyrnin About.com PC Hardware / Reviews Sign up for My Newsletter My Bio Headlines Browse Topic Find the Right PC Laptop PCs Desktop PCs PC Hardware Basics DIY Tutorials Monitors CD/DVD Drives Video Cards Hard Drives PC Audio Memory Processors Motherboards Cases PSUs and Cooling PC Gaming Product Reviews Alienware About.com Rating: ASUS ROG G750JN About.com Rating CyberpowerPC Fangbook EVO HX7 About.com Rating: Acer Aspire V G About.com Rating: Tablet Buyer's Guide. Tablets have become one of the fastest selling computing devices in history. Their compact size and low costs make them appealing to many people compared to laptops. Their media capabilities in particular are	71	7	71	7
http://scienceline.ucsb.edu/getter.php?ker=395	UCSB Science Line sctest	64	6	64	6
http://www.howstuffworks.com/pc.htm%23	HowStuffWorks "How PCs Work"	64	6	64	6
http://answers.yahoo.com/question/index?qid=20061009072639...	What is a computer? - Yahoo! Answers	32	3	32	3
http://answers.yahoo.com/question/index?qid=20131205024651...	What is a computer???? - Yahoo! Answers	57	6	25	3
http://www.brainpop.com/theChinlool/computersandInternet/Comp...	BrainPOP Computer Q - BrainPOP - Animated Educational ...	60	6	0	<1
http://www.raunamiske.noise.com/forum/topic18933-993-390...	Opis računalnika - Modifikacije Hlaenje navijanje ...	59	5	50	5
http://uk.answers.yahoo.com/question/index?qid=201312050346...	What is a computer????? - Yahoo! Answers UK	57	6	0	<1
http://golfski.wordpress.com/2011/12/14/about-computers/	About Computers GOLEKI Gujrat (Pakistan)	63	6	0	<1
http://remisaccessories.wordpress.com/2014/02/09/computer/	Computer remisaccessories	21	2	0	<1
http://www.factmonster.com/ipa/A0774698.html	How Do Computers Work? FactMonster.com	64	6	0	<1

Original Document

Za testiranje plagiatorstva, sem se odločil, da opise zgradbo in delovanje računalnika. Nekaj stavkov bom napisal z svojimi besedami, ostalo vključeno gradivo, pa bom pobral z interneta. To besedilo bom kasneje preverjal z programi za odkrivanje plagiatorstva in si njihove rezultate zabeležil, ter jih primerjal med seboj.

Računalnik je splošno uporabna naprava, ki se jo lahko programira za izvajanje aritmetičnih in logičnih operacij. Ker se zaporedja izvajanja operacij zlahka spremenijo, lahko računalnik rešuje različne vrste problemov.

Leta 1970 so naredili Imlac PDS-1, prvi računalnik, v katerem bi danes takoj prepoznali osebni računalnik, saj je imel svoj monitor (z vektorskim, ne rastrskim zaslonom, kakršne imamo danes) in tipkovnico, kar za računalnike tistega časa še zdaleč ni bilo kar samo po sebi razumljivo. Sodil je v

Found Text (<http://compreviews.about.com/>)

PC Hardware and Reviews - Computer Reviews and Information on Desktops, Laptops and Components Computing PC Hardware Reviews Search PC Hardware / Reviews Buyer's Guides PC Reviews Build A PC Share Must Reads What PC Should I Buy Tablet Buyer's Guide Laptop PC Buyer's Guide Desktop PC Buyer's Guide Best Chromebooks Guide since 2002 Mark Kyrnin About.com PC Hardware / Reviews Sign up for My Newsletter My Bio Headlines Browse Topic Find the Right PC Laptop PCs Desktop PCs PC Hardware Basics DIY Tutorials Monitors CD/DVD Drives Video Cards Hard Drives PC Audio Memory Processors Motherboards Cases PSUs and Cooling PC Gaming Product Reviews Alienware About.com Rating: ASUS ROG G750JN About.com Rating CyberpowerPC Fangbook EVO HX7 About.com Rating: Acer Aspire V G About.com Rating: Tablet Buyer's Guide. **Tablets have become one of the fastest selling computing devices in history. Their compact size and low costs make them appealing to many people compared to laptops. Their media capabilities in particular are**

Slika 20: Viper poročilo o plagiatorstvu.

S programom lahko tudi preverimo, če je določeno literarno delo že bilo objavljeno na spletnem medmrežju oziroma za preverjanje avtorstva besedila. Ker je enostaven za uporabo, je primeren tudi za začetnike pri uporabi tovrstnih programov. Ostalim pa prihrani čas in težave, ki jih povzročijo kompleksni programi. Z njim lahko pregledujemo tekstovne datoteke, pisarniške datoteke (npr. .doc, .docx), standardne datoteke za izmenjavo elektronskih dokumentov (.pdf) in datoteke, ki uporabljajo označevalni jezik za oblikovanje spletnih strani (.html).

Na voljo nam je tudi možnost, da se rezultati iskanja izvozijo in prikažejo kot poročilo v novi datoteki. V tem primeru nam program kreira datoteko .html, ki vsebuje podatke, ki smo jih pregledovali. Kjer program meni, da gre za plagiatorstvo, se nam ozadje besedila obarva z rdečo barvo ter tako pripomore k boljši preglednosti. Ali gre resnično za plagiatorstvo, pa mora preučiti človek.

Povem lahko, da me je program najbolj razočaral, ker ne podpira šumnikov. Šumnike napačno obravnava, zato je posledično tudi rezultatov pri preverjanju plagiatorstva veliko manj, kot bi jih moralo biti. Upam, da bodo v prihodnje to pomanjkljivost odpravili.

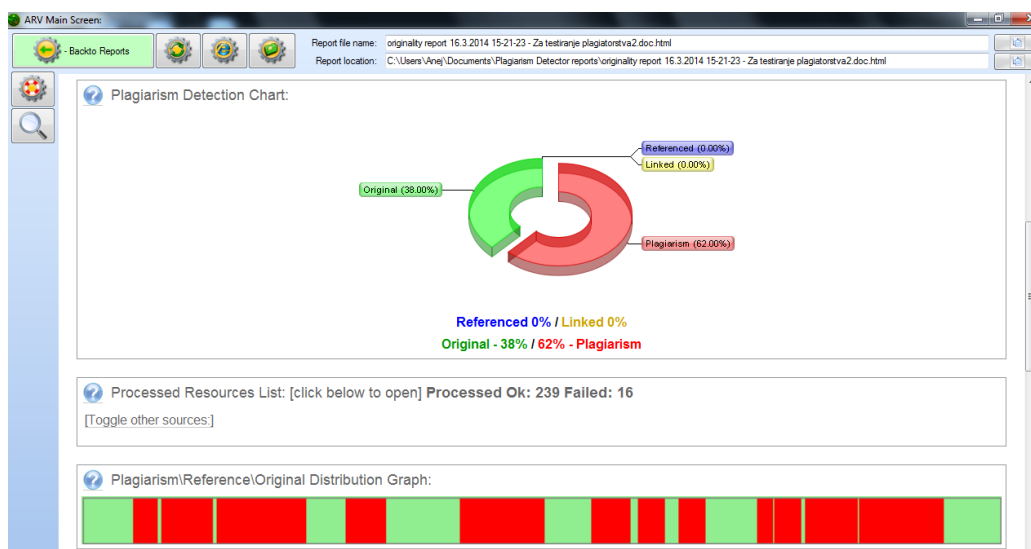
5.1.5 Plagiarism Detector

Glavna naloga programa je avtomatizirano odkrivanje digitalnega plagiatorstva (kot je nedovoljeno »copy-paste« iz tekstovnega gradiva), ki izvira iz svetovnega spleta s pomočjo iskalnikov kot so Google, Yahoo in AltaVista. Uporablja največjo digitalno javno podprto bazo podatkov na svetu, ki vključuje več kot sedem milijard spletnih virov.

Plagiarism Detector [20] je program, ki deluje v okolju Microsoft Windows (XP, Vista, Windows 7, Windows 8) in se uporablja za pregled sklopov besedila v dokumentih. Na podlagi iskalnih algoritmov pregleda bazo podatkov ter poskuša najti ujemajoče se vire ter opredeliti natančno število ujemajočih se informacij.

Za razliko od drugih spletnih storitev je to namizna aplikacija, ki ima edinstvene lastnosti. Ker nima strogih omejitev sistemskih virov, lahko pri preverjanju teksta uporabi precej procesorske moči in tako lahko hkrati iskalnikom pošljemo več zahtevkov. Tako izkoristi CPU moč med primerjanjem besedila in informacijami, ki se pošiljajo na internet, saj generira masovne poizvedbe in beleži zunanje vire, ki se ujemajo.

Program nam daje več možnosti pri preverjanju datoteke o vsebovanosti plagiatorstva. Prva možnost je, da preverjamo datoteko po svetovnem spletu s pomočjo iskalnih algoritmov. Kot drugo možnost imamo na voljo, da datoteko primerjamo z drugo datoteko, ki jo imamo shranjeno na našem računalniku (trdi disk, USB ključek itd.). Tretja možnost pa je, da določimo mapo, ki vsebuje datoteke z besedilom in naša datoteka se bo primerjala z njimi. V tem primeru se bodo pregledale tudi podmape in datoteke, ki jih podmape hranijo v izbrani mapi. V vseh primerih se nam ustvari poročilo o vsebovanosti plagiatorstva. Med drugim je v poročilu vsebovan tudi tortni grafikon (Slika 21), ki nazorno prikaže, koliko je avtorskega besedila, koliko besedila je bilo prekopiranega ter koliko besedila se nanaša na druga avtorska dela (citiranje). Pod njim je na grafikonu z rdečo barvo prikazano, kje se v besedilu pojavljajo prekopirani stavki ali zaporedje besed, glede na celoten tekst, ki smo ga pregledovali.



Slika 21: Tortni prikaz plagiatorstva, citiranja in našega lastnega besedila.

Program nam grafično izpiše tudi tri vire, iz katerih je bilo prepisanih največ stavkov ali besed, na koncu pa še analizira celotno besedilo. Analiza je prikazana tako, da se izpiše naše besedilo, kjer pa so stavki ali več besed, ki se štejejo za plagiat, se prikažejo z rdečo podlago (Slika 22). Pod njimi je spisek treh URL povezav, ki nas ob kliku nanje preusmerijo na spletno stran, kjer se pojavljajo enaki stavki ali besede. Če smo veliko citirali, se prikaže še spisek URL povezav, na katere se nanaša večina citiranja.

Top 3 Plagiarized Sources:

Chars#	Words#	URL
559	92	http://www.djaski.net/gettrif_plo_racunalniki_informatika_zgodovina_racunalnikov_01.doc
559	92	http://www.djaski.net/gettrif_plo_racunalniki_informatika_zgodovina_racunalnikov_01.odt
526	92	http://colos1.fri.uni-lj.si/ER/INFORMATIKA/informacijska_tehnologija/zgradba_racunalnika.html

Top 3 Referenced Sources:

Chars#	Words#	URL
323	48	http://tehnologija.wordpress.com/vrste-racunalnikov
154	23	http://sl.wikipedia.org/wiki/Ra%C4%B0unalnik
151	23	http://prezi.com/fkyzbtbikxp5/racunalniki

Detailed Document Analysis:

Za testiranje plagiatorstva, sem se odločil, da opišem zgradbo in delovanje računalnika. Nekaj stavkov bom napisal z svojimi besedami, ostalo vključeno gradivo, pa bom pobral z internetnega omrežja. To besedilo bom kasneje preverjal z programi za odkrivanje plagiatorstva in si njihove rezultate zabeležil, ter jih primerjal med seboj. Računalnik je splošno uporabna naprava, ki se jo lahko programira za izvajanje aritmetičnih in logičnih operacij. Ker se zaporedja izvajanja operacij zlahka spremenijo, lahko računalnik rešuje različne vrste problemov. Leta 1970 so naredili Imlac PDS-1, prvi računalnik, v katerem bi dandanes takoj prepoznali osebni računalnik, saj je imel svoj monitor (z vektorskim zaslonom, kakršne imamo dandanes) in tipkovnico, kar za računalnike tistega časa še zdaleč ni bilo kar samo po sebi razumljivo. Sodil je v razred grafičnih miniračunalnikov in je bil še veliko predrag, da bi mu lahko rekli osebni računalnik. Računalnik je naprava za shranjevanje in obdelavo podatkov in ena od naprav za prenos informacij. Podatek predstavlja dejstva, pojme na formaliziran način, s simboli. Informacija so ovrednoteni podatki v specifični situaciji (G.C.Everest, 1986). Podatki niso informacije, podatki posredujejo informacijo prejemniku, katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta, na katerega se nanašajo, če je količina podatkov tako velika, da se jih v času, ki je na voljo za ukrepanje na njihovi osnovi, ne da interpretirati, se lahko zgodi, da s podatki ni posredovana nobena informacija. (B. Langefors, 1980) Računalniki ali tudi računalniški sistemi so stroji/avtomati z obdelavo informacij sestavljeni iz strojne opreme, programske opreme in

Slika 22: Prikaz, na katerih URL naslovih se pojavlja identično besedilo, kot ga imamo mi v tekstu.

Tukaj bi še omenil, da program omogoča preverjanje za večino tekstovnih datotek (.doc, .docx, .txt), predstavitevni datotek (.ppt, .pptx), standardnih datotek za izmenjavo elektronskih dokumentov (.pdf) in datotek, ki uporabljajo označevalni jezik za oblikovanje spletnih strani (.html, .php). Program je plačljiv, vendar imamo možnost namestitve poskusne različice, ki jo lahko zaženemo največ desetkrat, nadaljnja preverjanja pa so plačljiva.

5.1.6 Primerjava rezultatov plagiatorstva med storitvami ter programi

Pri testiranju programov in storitev sem prišel do presenetljivih rezultatov. Sprva sem mislil, da bodo rezultati povsod enaki ali vsaj podobni. Vendar se kar precej razlikujejo med seboj. Razlog za to je v tem, da vsaka storitev ali program uporablja pri svojem izvajanju drugačno bazo podatkov ter drugačne iskalne algoritme. Zato pridemo do takih razlik, ki se kažejo v učinkovitosti iskanja plagiatorstva (Tabela 2).

Orodje	Čas izvajanja (sekunde) za datoteke, ki vsebujejo 1000 črk			Podprti formati	Plagia - torstvo	Plačljiv
	Datoteka (.doc)	Datoteka (.txt)	Vnos besedila		%	
Plagiarism Check.org	32	20	18	.xls, .xlsx, .xlsm, .xlsb, .xltx, .xltm, .xlv, .csv, .dbf, .ods, .doc, .docx, .txt, .rtf, .dotx, .dot, .html, .htm, .xps, .mhtml, .mhtml, .xml, .wps, .odt, .ppt, .pptx, .pot, .potx, .pps, .ppsx, .odp	25	ne
Small SEO Tools	/	/	375	prosti tekst	54	ne
PlagScan	50	37	32	.docx, .doc, .pdf, .txt, .html, .doc6, .doc95, .wps, .wpd, .odt, .ott, .rtf, .sdw,	45	da – 9,99 €

Orodje	Čas izvajanja (sekunde) za datoteke, ki vsebujejo 1000 črk			Podprti formati	Plagia - torstvo	Plačljiv
	Datoteka (.doc)	Datoteka (.txt)	Vnos besedila		%	
				.sdw4, .sdw3, .sxw		
Viper	200	66	/	.doc, .docx, .pdf, .html, .odt, .rtf, .text, .s, .cs, .app, .java, .ptt, .pttx	32	ne
Plagiarism Detector	306	48	/	.doc, .docx, .pdf, .rtf, .html, .htm, .php, .asp, .odt, .dotx, .ppt, .pptx, .txt	62	da – od 22 € naprej

Tabela 2: Primerjava programov in storitev za odkrivanje plagiatorstva v prostem tekstu.

Kot lahko iz Tabele 2 razberemo, se čas izvajanja (preverjanja) plagiatorstva ne razlikuje precej. Včasih je veljalo pravilo, da so bili programi precej hitrejši od spletnih storitev, saj ti s pomočjo računalniškega procesorja pošiljajo dele kode (besede) spletnim brskalnikom, ki pregledajo po internetu, če se besede ujemajo. Hitrost izvajanja je bila odvisna tudi od hitrosti procesorja, saj je zmogljivejši lahko poslal več besed v enakem časovnem obdobju kot drugi. Sedaj pa ni več tako velikih razlik, saj imajo strežniki, ki jih uporabljajo spletne storitve za pregledovanje plagiatorstva močnejše procesorje, kot jih imajo navadni uporabniki in izvedejo vse potrebne operacije v najkrajšem času.

Pri programih in storitvah, ki omogočajo pregled pisarniških datotek (npr. .doc, .docx), standardnih datotek za izmenjavo elektronskih dokumentov (.pdf) in še nekaterih, se čas izvajanja upočasni, saj uporabljajo drugačne zapise kot zgolj navadno besedilo. Zato je potrebno iz take datoteke najprej povzeti besedilo ter ga pretvoriti v navaden tekst, saj je šele nato mogoče izvajati operacije z njim.

Pri programu Viper vidim kot glavno slabost to, da ne podpira šumnikov, zato je tudi rezultat plagiatorstva izredno majhen. Za literaturo v tujem (npr. angleškem) jeziku pa je popolnoma enakovreden Plagiarism Detectorju.

Pri plačljivih programih ali storitvah je cena kar visoka (od 130€ naprej) in si jih ne more privoščiti ravno vsak. Zato se licence podeljujejo predvsem organizacijam, univerzam, inštitutom ...

5.1.7 Priporočilo

Po vseh testiranjih, ki sem jih izvedel, bi za brezplačno preverjanje plagiatorstva priporočal uporabo storitve Plagiarism check.org, saj omogoča preverjanje za neomejeno število dokumentov. Za uporabo je potrebna zgolj registracija uporabnika, s tem pa pridobimo še prostor na strežniku, kjer se nam hranijo pregledane datoteke. Tako imamo rezultate vedno na razpolago na strežniku, lahko pa si jih prenesemo tudi na svoj računalnik v obliki .pdf datotek. Storitev je zelo dobro zasnovana, bi bilo pa zelo zaželeno, da bi se ji dodali novi iskalni algoritmi, saj za slovenski jezik ni ravno najboljša izbira. Za angleški jezik pa lahko zagotovim, da je izredno zanesljiva storitev, saj so rezultati nazorno prikazani ter povsem primerljivi z drugimi plačljivimi orodji.

Za plačljivo preverjanje plagiatorstva bi priporočal uporabo programa Plagiarism detector. Pred uporabo je potrebna namestitev programa na lokalni računalnik. Pri odkrivanju plagiatorstva si program pomaga s pomočjo iskalnikov, kot so Google, Yahoo in AltaVista ter tako uporablja eno največjih digitalno javno podprtih baz podatkov na svetu. Zato so tudi rezultati, ki jih pridobi, izredno natančni in zanesljivi. Prikaz rezultatov o plagiatorstvu prikaže tudi grafično v obliki tortnega prikaza ter tako hitro vidimo, koliko je bilo lastnega besedila, koliko citiranja ter koliko je bilo teksta, kjer se je odkrilo podvajanje na medmrežju. Podvajanje, ki ga je program odkril, nam nazorno prikaže, obenem pa nam izpiše, na katerih URL naslovih se pojavlja identično besedilo, kot ga imamo sami v tekstu. Ta program ima tudi možnost, da pregleda naš dokument in ga primerja z dokumenti, ki jih imamo shranjene v mapah in podmapah na lokalnem računalniku ali pa s poljubno izbrano datoteko. Cena licence za uporabo tega programa je nekaj več kot 40 € in lahko potrdim, da je program vreden tega denarja.

6 Zaključek

V diplomskem delu sem najprej raziskoval, kako obširen pojem je pravzaprav plagiatorstvo, kdaj se je pojavilo, kje vse se z njim srečujemo, kakšne posledice prinaša ter kako ga omejiti. Ugotovil sem, da je plagiatorstvo tako razširjeno, da ga je praktično nemogoče ustaviti in se žal tudi na vseh področjih hitro povečuje. Posledica tega je, da je čedalje manj ljudi, ki so pripravljeni samostojno ustvarjati in se za izhodišče raje zanašajo na že napisana dela.

Nato sem se poglobil v delovanje orodij za odkrivanje plagiatorstva v programski kodi in prostem tekstu. Preden je preverjanje postalo avtomatizirano, se je izvajalo ročno in enakosti se je le stežka odkrilo, zato so bili plagiati večinoma prikriti. Z avtomatiziranim preverjanjem pa se plagiatorstvo lahko vsaj deloma omeji ter omogoča odkritje več plagiatorjev.

Pri odkrivanju plagiatorstva v programski kodi sem ugotovil, da so orodja zelo koristen pripomoček za delavce v izobraževalnih ustanovah, čeprav ne zagotavljajo zadostne uspešnosti za boj proti plagiatorstvu. Tako preverjanje se lahko samodejno izvede že, ko študent pošlje svojo nalogo na ocenjevanje na strežnik izobraževalne ustanove. Tako profesorju prihrani veliko časa, saj mu ni potrebno posebej ugotavljati, če gre za prepisovanje, temveč se lahko osredotoči zgolj na pravilnost delovanja programske kode. Nekatera orodja omogočajo preverjanje plagiatorstva za več programskih jezikov, zato izobraževalnim ustanovam ni potrebno preučevati več orodij, ampak se osredotočijo na uporabo enega. Kot naslednjo prednost pa bi izpostavil orodja, ki imajo možnost razširitve ali izboljšanja programa, saj lahko na ta način dopolnimo orodje, ki nam bo najbolj ustrezalo v dani situaciji.

Pri odkrivanju plagiatorstva v prostem tekstu imamo na razpolago veliko izbiro programov in spletnih storitev. Ti so lahko pripomoček zaposlenim v izobraževalnih ustanovah, ki preverjajo enakost literarnih del (tudi nekatera orodja za odkrivanje plagiatorstva v programski kodi imajo na razpolago možnost odkrivanja plagiatorstva v prostem tekstu, vendar ta orodja preverjajo besedilo tekstovnih datotek med seboj zgolj lokalno) ali uporabnikom, ki želijo preveriti, če je določeno literarno delo že bilo objavljeno na medmrežju oziroma za preverjanje avtorstva besedila. Ponavadi se ujemanje besedila preverja s pomočjo spletnega medmrežja, pri nekaterih pripomočkih pa lahko tudi lokalno. Pri ugotavljanju podobnosti se ni mogoče v celoti zanesti na različne računalniške programe za odkrivanje plagiatov, saj ti programi razpolagajo z omejenim številom podatkovnih baz, ki ne vsebujejo vseh besedil. Pri tem lahko gre tudi za prevod tujega besedila v domač jezik, česar programi za odkrivanje plagiatorstva ne ugotovijo. Zato se kljub nenehnemu izpopolnjevanju njihovih iskalnih algoritmov ne moremo nanje z gotovostjo zanesti.

Zaradi hitrega napredka v tehnologiji interneta imamo na razpolago ogromne količine informacij. Te informacije si nato izmenjujemo med seboj in institucijami, kar velikokrat privede do tega, da osebe pošiljajo enake informacije in velikokrat pride do podvajanja. Zato ima plagiatorstvo vse možnosti, da spodkoplje tradicionalno izdelavo nalog in drugih izdelkov, ki jih mora učenec ali študent izdelati v učnem procesu. Za preprečevanje tega bi morali v izobraževalnih ustanovah posvetiti več pozornosti ozaveščanju šolarjev, dijakov ali študentov o prednostih ter pomenu lastnega intelektualnega dela.

Literatura in viri

Literatura:

- [1] A. Parker, J. Hamblen, »Computer Algorithms for Plagiarism Detection«, *IEEE Transactions on Education*, str. 94–99, 1989.
- [2] B. Kaučič, D. Sraka, M. Krašna, »Plagiarizem pri poučevanju programiranja« *Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference*, Portorož, Slovenija, okt. 2008, str. 325–328.
- [3] D. Wedam Lukić, »O plagiatu, intelektualni poštenosti ter etiki znanstvenega in strokovnega pisanja«, *Podjetje in delo*, str. 984–988, 2013.
- [4] H. Jenull, »Kazenskopravni vidiki plagiata«, *Podjetje in delo*, str. 1011–1026, 2013.
- [5] M. Rosita Schiller, »E-Cheating: Electronic Plagiarism«, *Journal of the American Dietetic Association*, str. 1058–1062, 2005.
- [6] M. Trampuž, »Avtorskopravni vidiki plagiata s poudarkom na pisanih znanstvenih delih«, *Podjetje in delo*, str. 989–1003, 2013.

Viri:

- [7] Algoritem za delovanje orodja JPlag (2000). Dostopno na: <http://page.mi.fu-berlin.de/prechelt/Biblio/jplagTR.pdf>.
- [8] Algoritem za delovanje orodja SIM (2002). Dostopno na: <http://www.eng.uwi.tt/depts/elec/staff/feisal/ee302/sim-gitchell.pdf>.
- [9] Algoritem za delovanje sistema Moss (2003). Dostopno na: <http://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>.
- [10] Citiranje in navajanje virov (2006). Dostopno na: http://www.tre.sik.si/main/pomoc/files/citiranje_in_navajanje_virov.pdf.
- [11] Digitalna knjižnica Maribor (2014). Dostopno na: <http://dkum2.uni-mb.si/podrocje.aspx?id=0>.
- [12] Marble programsko odkrivanje plagiatorstva (2007). Dostopno na: <http://www.cs.uu.nl/wiki/pub/Hage/ResearchTalks/marble-slides.pdf>.
- [13] Obširnejši opis plagiatorstva (2014). Dostopno na: <http://www.plagiarism.org/plagiarism-101/what-is-plagiarism>.
- [14] Odkrivanje plagiatorstva (2011). Dostopno na: <http://www.biblioblog.si/2011/03/plagiatorstvo-in-njegovo-odkrivanje.html>.

- [15] Odkrivanje plagiatorstva v programski kodi (2011). Dostopno na: <http://my.safaribooksonline.com/book/networking/security/9780137035816/software-plagiarism-detection/ch13lev1sec1>.
- [16] Opis avtorskih pravic (2014). Dostopno na: <http://www.intelektualna-lastnina.si/avtorska-in-sorodne-pravice/avtorska-pravica>.
- [17] Opis delovanja programov pri odkrivanju plagiatorstva (2004). Dostopno na: <http://www1.spms.ntu.edu.sg/~chenxin/paper/IT04a.pdf>.
- [18] Opis intelektualne lastnine (2014). Dostopno na: <http://www.pravniportal.si/#!/intelektualna-lastnina/cawa>.
- [19] Opis plagiatorstva (2014). Dostopno na: <http://www.lib.usm.edu/legacy/plag/whatisplag.php>.
- [20] Opis programa Plagiarism detector (2014). Dostopno na: <http://www.plagiarism-detector.com/>.
- [21] Orodja za odkrivanje enakosti v programski kodi (2012). Dostopno na: <https://files.semtech.athabascau.ca/public/TRs/TR-SemTech-07022012.pdf>.
- [22] Orodja za odkrivanje plagiatorstva v programski kodi (2011). Dostopno na: http://www.ou.nl/Docs/Expertise/OTEC/Nieuws/cserc2011/presentation_jurriaan_hage.pdf.
- [23] Orodja za preverjanje plagiatorstva (2010). Dostopno na: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.1043&rep=rep1&type=pdf>.
- [24] Orodje JPlag (2014). Dostopno na: <http://jplag.ipd.kit.edu/>.
- [25] Orodje Sim (2014). Dostopno na: http://dickgrune.com/Programs/similarity_tester/.
- [26] Plagiatorstvo je smrt kreativnosti (2012). Dostopno na: <http://chuppacadabra.com/zanimivosti/plagiatorstvo-in-copy-paste-smrt-kreativnosti/>.
- [27] Plagiatorstvo opis (2013). Dostopno na: <http://www.student.si/preberisi/aktualno/plagiat.html>.
- [28] Plagiatorstvo v diplomskih nalogah (2014). Dostopno na: <http://filternet.si/dd/clanki/kdaj-je-diploma-plagiat/>.
- [29] PlagScan spletna storitev za odkrivanje plagiatorstva (2014). Dostopno na: <http://www.plagscan.com/>.
- [30] Preverjanje plagiatorstva ne bo nikoli popolno (2013). Dostopno na: <http://www.delo.si/druzba/znanost/preverjanje-plagiatorstva-ne-bo-nikoli-popolno.html>.
- [31] Storitve Plagiarism CHECK.org (2014). Dostopno na: <http://plagiarismcheck.org/>.
- [32] Program Viper (2014). Dostopno na: <http://www.scanmyessay.com/>.
- [33] Repozitorij Univerze v Ljubljani (2014). Dostopno na: <http://rul-test.uni-lj.si/info/index.php/slo/kaj-je-rul>.

- [34] Rezultati neodvisnih testiranj spletnih storitev za odkrivanje plagiatorstva (2013). Dostopno na: <https://www.plagscan.com/test-reports>.
- [35] Sistem Moss (2011). Dostopno na: <http://theory.stanford.edu/~aiken/moss/>.
- [36] SSKJ: plagiat (2014). Dostopno na: http://bos.zrc-sazu.si/cgi/a03.exe?name=sskj_testa&expression=plagiat&hs=1.
- [37] Storitve Small SEO Tools odkrivanje plagiatorstva (2014). Dostopno na: <http://smallseotools.com/plagiarism-checker/>.
- [38] Še nekaj o plagiatorstvu (2014), ter kako se mu izognemo. Dostopno na: <http://www.pedagogika-andragogika.com/plagiatorstvo>.
- [39] Še nekaj podatkov o plagiatorstvu (2014). Dostopno na: http://sfhomeless.wikia.com/wiki/San_Francisco_Homeless_Resource:Plagiarism.
- [40] Tuji slovarji obrazložitev besede plagiatorstvo (2014). Dostopno na: <http://dictionary.reference.com/browse/plagiarism>.
- [41] Uveljavljanje pravic intelektualne lastnine (2014). Dostopno na: <http://www.uil-sipo.si/uil/urad/o-intelektualni-lastnini/uveljavljanje-pravic/ukrepi-pri-krsitvah/>.
- [42] Zgodovina avtorskih pravic (2013). Dostopno na: http://wikieducator.org/Copyright_for_Educators/History.
- [43] Zgodovina plagiatorstva (2009). Dostopno na: <http://hnn.us/article/569>.