Autor: Zoran Hercigonja, mag.edu.inf.

RAČUNALNA DETEKCIJA PLAGIJATA

Pregled metoda i algoritama

ISBN 978-953-59549-7-2

Impressum

Naslov: **Primjeri pisanih priprema za izvedbu nastavnog sata iz metodike nastave informatike za osnovne i srednje škole**
Autor: **Zoran Hercigonja, mag.edu.inf.**

Lektor: **Adela Brozd**
Nakladnik: **Vlastita naklada autora**
Mjesto i godina izdavanja: **Imbriovec Jalžabetski 45c, 2017. godine**
Vrsta publikacije: **Stručna elektronička knjiga**
Medij: **Mrežna publikacija**
Format: **A4**
Br. stranica: **36**
Područje: **Računalstvo-detekcija plagijata**
ISBN 978-953-59549-7-2

Sadržaj:

1. Osnovno o plagijarizmu

Iznimno je važno razlučiti pojam plagijarizma od ostalih pojmova s kojima on ulazi u diskrepanciju i sinonimnu vezu. Najčešće se s pojmom plagijarizma povezuje i pojam prepravljanja te kompilacije koji imaju sasvim drugačije značenje od konteksta u kojem shvaćamo pojam plagijarizma. Da bi se smanjile nejasnoće i miješanje pojmova, date su jasne i jednoznačne definicije plagijarizma.

1.1.Pojam plagijarizma

Većina autora konzistentno definira pojam plagijarizma kao preuzimanje tuđih ideja, postupaka ili sadržaja u obliku teksta bez navođenja izvora, bez referenciranja ili parafraziranja. Za razliku od njih, Hrannabuss (2001) je precizirao pojam plagijarizma na „...neovlašteno korištenje ili vrlo bliska imitacija ideja, jezika, tema i predmeta...“. Tom definicijom se daje do znanja da plagiranje nije samo preuzimanje sadržaja bez navođenja nego i preuzimanje stila pisanja ili prilagođavanje stila pisanja koji je potekao od autorovog originalnog djela. Iako se pojam plagijarizam izvodi iz latinskog „plagere“ što u prijevodu znači čin prisvajanja ili kopiranja tuđeg rada u vlastiti u obliku cjeline ili odlomaka, mora se naglasiti da se ne radi samo o sadržaju ili frazama, već i o imitaciji same ideje i jezika kojim je ta ideja razrađena i realizirana. Dakle čin plagiranja podrazumijeva nešto više od obične kompilacije ili prepisivanja činjenica koje se dovode u zajedničku vezu kao pojmovi ili sinonimnu vezu kao jednoznačni termini. Prisvojen stil pisanja omogućava plagijatoru dodatno prepravljanje to jest dodavanje ili modificiranje postojećeg sadržaja u duhu uobličenog stila pisanja specifičnog pojedinom autoru. Zbog toga se spomenuti pojam kompilacija ne može poistovjetiti u potpunosti s plagiranjem. Kompilacija je dio plagiranja i sastoji se u prepisivanju činjenica to jest doslovnom (mehaničkom) prepisivanju i prenošenju riječi, fraza i odlomaka.

Još jedna vrlo dobra iako ne toliko precizna definicija koja objedinjuje većinu elemenata (ideja, jezik, tema, predmet) pojma plagijarizam je Samuelsonova definicija plagijarizma „...Ono što je pogrešno u plagijarizmu i plagijatima nalazimo u činjenici da neka osoba za plagirani tekst/djelo tvrdi da je njezino, iako jako dobro zna da je nastalo iz drugog izvora, pretpostavljajući da čitatelj to neće znati i nadajući se da će imati koristi od čitateljeva neznanja...“ (Samuelson, 1994). Dakle plagijat je preuzeta sveukupnost stila, jezika, ideje i konkretnih činjenica koje su preuzete namjerno ili nenamjerno tako da „...osoba za plagirani tekst/djelo tvrdi da je njezino, iako jako dobro zna da je nastalo iz drugog izvora...“ (Samuelson, 1994). Iako je ta definicija na prvi pogled „sveobuhvatna“ nije dovoljno precizna poput Hrannabussove definicije. U (Hrannabuss, 2001) se jasno kaže „...neovlašteno korištenje ili vrlo bliska imitacija ideja, jezika, tema i predmeta...“ dok kod (Samuelson, 1994) se samo kaže „...osoba za plagirani tekst/djelo tvrdi da je njezino, iako jako dobro zna da je nastalo iz drugog izvora...“. Hrannabussova definicija precizno ukazuje na glavne elemente koji ulaze u pojam plagijarizma i ne dovodi do pogrešnih zaključaka. Primjerice zbog preopćenitosti Samuelsonove definicije pojedinac bi pojam plagijarizma mogao poistovjetiti s kompilacijom ili prepravljanjem. To se jasno vidi u „...iako jako dobro zna da je nastalo iz drugog izvora...“ (Samuelson, 1994) čime se aludira na mogućnost mehaničkog prepisivanja činjenica te mogućnost modifikacije izvornog sadržaja dodavanjem ili ispuštanjem činjeničnih informacija iako je jako dobro poznato plagijatoru da su te modificirane informacije nastale iz originalnog izvora. Prema tome Hrannabussovu definiciju pojma plagijarizam možemo smatrati potpunom definicijom plagijarizma. Spomenuto je da je plagijat vrlo usko povezan s pojmom prepravljanja. Pojmovi plagiranja i prepravljanja nisu sinonimi i nije ih moguće smatrati jednim pojmom, jer prepravljanje ne mora podrazumijevati prepravljanje stila ili jezika već samo činjenica dodavanjem ili izmišljanjem fraza. Prema (Baždarić i sur, 2009), pojam prepravljanja se definira kao „...naknadno mijenjanje ili izostavljanje postupaka ili rezultata kako bi se objavio željeni ili izbjegao neželjeni rezultat...“. Upravo to mijenjanje ili izostavljanje činjenica ne mora značiti nužno plagiranje, jer sofisticirano plagiranje znači i izmjenu jezika, prilagodbu stila i prepisivanje činjenica. Dakle preuzimanje nečije ideje znači „...neovlaštena uporaba jezika i misli drugih autora i predstavljanje ih kao vlastitih...“ (Joy i Luck, 1999). Prema tome plagijat nije samo prepisivanje činjenica nego i prisvajanje jezika i stila pisanja.

Kompilaciju i prepravljanje možemo zato smatrati fazama plagiranja, ali ne i plagiranjem to jest preuzimanjem činjenica, stila i jezika. Zato nije ispravno smatrati plagijat radom nastalim uključivanjem dijela ili cjeline sadržaja druge osobe bez navođenja originalnog autora. Navođenje autora i prepisivanje činjenice je jedna stvar, no ona bitnija je stil i jezik pisanja. Plagijat je iznimno kompleksna kategorija koja se može identificirati kroz svoje pojavne oblike koji pružaju pomoć pojedincima u identifikaciji i traženju sličnosti u sadržaju, formi, stilu ili jeziku pisanja s obzirom na pojavni oblik.

1.2.Pojavni oblici plagijarizma

Plagijati se pojavljuju u velikom broju pojavnih oblika. Klasifikacije pojavnih oblika variraju od autora do autora pa je teško sa sigurnošću utvrditi stvarni broj izvornih podjela kao i parametara njihove identifikacije. Stoga je navedeno nekoliko kategorija koje najbolje objedinjuju većinu ponuđenih klasifikacija raznih autora. U uvodnom djelu je identificirano namjerno i nenamjerno pojavljivanje plagijata. Spomenuto je da plagijati mogu nastati hotimičnim postupcima ili sasvim nehotimično odnosno slučajno. Hotimičnost bi svakako podržavala plagiranje s namjerom da se prevari čitatelja te da se određeni sadržaj prisvoji i podmetne kao vlastiti. Plagijati ne moraju nužno uvijek predstavljati krađu autorstva, već mogu predstavljati i autoplagijatorstvo.

Stoga prema (Beasley, 2006), postoje sljedeće vrste plagijata:

1. ***Slučajni plagijarizam***: zbog nedostatka znanja o plagijarizmu i vještina pravilnog citiranja i referenciranja izvora. Najveća opaska slučajnog plagijarizma se daje upravo neznanju i nepoznavanju pravilnika, odredaba i etičkih kodeksa koji propisuju odredbe i pravila o pisanju vlastitih radova, o citiranju ili navođenju autora uz pojedine fraze i citate. U ovu kategoriju bi se mogao ubrojiti i nemar to jest ne vođenje računa o povredi akademske časti, ali i primjerice uključivanje pojmova i fraza u vlastiti rad koje su zbog učestalosti primjene u svakodnevnom jeziku struke, doslovno srasle s osjećajem posjedovanja i originalnosti. Današnje doba informacija je toliko opterećeno dnevnom bazom fraza i odlomaka koji postaju dio naše svakodnevice, nešto uobičajeno. Upravo takvo nešto uobičajeno postaje slučajan plagijat, jer nismo ni svjesni da učestalom uporabom definicija i fraza stvaramo utisak posjedovanja tih istih pojmova, fraza i definicija.

2. ***Nenamjerni plagijarizam*:** dostupnost različitih informacija utječe na naše misli te se iste ideje i izrazi mogu stvoriti u obliku izgovorenih i napisanih izraza različitih autora. Razdoblje informatizacije i krcatosti informacijama, navodi nas da unaprijed vlastite ideje i misli izražavamo frazama, terminologijom i oblicima koji su srodni onim napisanim ili izgovorenim izrazima autora. Ova vrsta plagijata je vrlo slična **slučajnim** plagijatima, jer u suštini u obje situacije zbog učestale izloženosti informacijama, kreiramo fraze i izražavamo ideje na način kako je to utkano u našu podsvijest, jer izloženost ponavljajućem nizu svakodnevnih informacija, a ponajviše informacija u struci upravo potiču osobu da se koristi i izražava na način na koji je to učinio autor određenog rada i djela. Ovo bismo mogli nazvati čak i podsvjesno nenamjerni plagijatom jer fraze i ideje koje su u neprestanom, ponavljajućem nizu emitirane potiču subjektivno prisvajanje bez osjećaja da postoji određeni izvor ili autor.
3. ***Namjerni plagijarizam***: namjerni čin kopiranja i prisvajanja, potpunog ili djelomičnog rada drugog autora bez naznačavanja originalnog izvora od kojeg se preuzima.

   Tu se podrazumijeva čisto hotimično odnosno namjerno preuzimanje sadržaja, stila i jezika kako bi se rad predstavio kao vlastiti bez navođenja autora. Naravno i u ovoj situaciji treba imati u vidu mogući nedostatak vještina kritičkog analiziranja i osvrta na autorove odlomke i fraze te nedostatak inicijative da se sadržaj interpretira na pravilan način. Ukoliko je osoba u deficitu sa sposobnosti kritičkog konzumiranja određenog sadržaja, ona će gotovo uvijek preuzeti dio i cjelinu sadržaja i predstaviti ga pod svoje, jer je u tom sadržaju pronašla takoreći izražaj vlastitih misli i ideja. Upravo to može navesti osobu da direktno koristi autorove sadržaje bez navođenja. Druga je situacija kada osoba pokušava prepravljati sadržaj to jest izostaviti određene dijelove ili izmišljati nove na temelju postojećeg sadržaja. Tu najčešće zbog izmišljanja i dodavanja ili izostavljanja sadržaja dolazi do neslaganja u stilu i jeziku kojim je pisano originalno djelo.

4. ***Autoplagijarizam*:** korištenje vlastitog, prije objavljenog rada u nekog drugom obliku, bez upućivanja na izvorni dokument. Ovaj oblik plagijata može se interpretirati situacijom u kojoj smo u novom radu pokušali koristiti fraze, odlomke ili cijela poglavlja koja smo ranije napisali u nekom drugom radu i djelu. Nedostatak referenciranja na prethodni sadržaj nekog izvornog dokumenta se ne interpretira kao ozbiljan oblik plagiranja, ali u tehničkom smislu se može shvatiti kao plagijat, jer ne predstavlja izvornost ideje i fraza. To nije toliko pogubno koliko ne referencirati se i pozivati na autore čije ideje, rečenice, fraze i terminologiju koristimo.

Sljedeća klasifikacija pojavnih oblika plagijata, odnosi se na već spomenutu „potpunu“ definiciju plagijata koja ubraja i predmet plagiranja, jezik , fraze i stil. Dakle prema (Hrannabuss S. 2001) plagijat je „...neovlašteno korištenje ili vrlo bliska imitacija ideja, jezika, tema i predmeta...“. Prema toj definiciji, plagijate je Olsson (2010) klasificirao kao:

1. ***Arheološki plagijati***: djelo je preuzeto s površinskom zamjenom i preraspodjelom pojedinih dijelova. Takav oblik plagijata podrazumijeva jednostavno izostavljanje ili dopunjavanje po potrebi kako bi se prikrio izvorni tekst ili sadržaj. Veći naglasak se stavlja na prepravljanje nego na pravo plagiranje jer se više-manje mijenja sadržaj na način da se izmišljaju dodatni dijelovi koji zamjenjuju postojeće ili se ispuštaju određeni dijelovi i fraze kako bi izvorni tekst bio što manje uočljiv te kako bi se što manje sumnjalo u originalnost i izvornost.
2. ***Dijakronijski plagijat:*** djelo je uzeto iz ranije razdoblja te se pokušava prikriti vrijeme njegova nastanka pretvarajući ga u djelo iz vlastitog vremena. Ovdje se jasno mogu potvrditi svi elementi Hrannabussove definicije plagijata. Kao osnovno drugi vremenski kontekst i vrijeme nastanka tog djela podrazumijeva i određeni stil pisanja koji je bio svojstven samo za to vrijeme, zatim jezik koji se upotrebljavao kao govorni jezik tog doba ili razdoblja. Uspješan plagijator bi sada trebao savladati jezik tog vremena i stil pisanja da bi mogao takvo djelo prevesti u djelu svojeg vremena. To je iznimno teško i vrlo će se često prikrasti koje kakve nedoumice zbog neslaganja među pojmovima ili će se javiti nelogičnosti zbog vrlo nevještog zamjenjivanja riječi onog vremena i vremena u kojem se trenutačno piše rad. Takav oblik plagijata plagijatora ostavlja vrlo ranjivim, jer samo vrlo sofisticirani i spretni plagijator može savladati i jezik i stil pisanja te ga prevesti u okvire u kojima on piše svoj rad.

Primjerice u takvom plagijatu nije dovoljno samo prepravljati jer dodavanjem ili izostavljanjem riječi ne dobiva se ništa. Stil i jezik tog vremena bit će previše uočljivi te će pobuditi sumnje u originalnost djela.

3. ***Kulturološki plagijat:*** podrazumijeva pokušaj preuzimanja tuđih kulturoloških artefakata i pretvaranje u artefakte vlastite kulture. Slično kao i za prethodni oblik plagijata i ovaj plagijat objedinjuje Hrannabussovu definiciju plagijata. Kulturološki elementi nekog djela su izazov za plagiranje. Kultura podrazumijeva i određeni jezični kontekst i uvriježeni stil pisanja kojeg je dosta teško oponašati. Isto tako premještanje kulturoloških artefakata znači i odgovarajuće razdaljine između pojmova jer u jednoj kulturi određeni pojam može imati drugačije značenje. Iznimno je teško plagirati u kulturološkome kontekstu. No ako se inzistira na činu plagiranja, najčešće dolazi do miješanja među pojmovima, zamjene pojmova ili pogrešne interpretacije. Na temelju toga je moguće s iznimnom lakoćom utvrditi plagijat. Pretvaranje kulturoloških artefakata u artefakte vlastite kulture podrazumijeva iznimno dobro savladavanje jezika i kulture pisanja te iznimno zahtjevno manipulaciju pojmovima odnosno prepravljanje.

Budući da je cilj ovog rada dati pregled dodataka za Moodle sustav koji ostvaruju aktivnost detekcije plagijata, najčešći tipovi plagijata koji su identificirani u području primjene softvera za automatsko detektiranje plagijata dijele se u pet kategorija.

Pojavni oblici plagijata koje navode (Weber-Wullf i sur, 2013) su:

1. ***Kopiraj/zalijepi plagijati (Copy & Paste):*** to su više-manje jedina vrsta lako prepoznatljivih plagijata. Plagijator kopira velik dio sadržaja ili odlomke predstavljajući ih pod svojim imenom. Povremeno plagijator preuzima cijeli rad i dopisuje samo svoje ime. Kopiraj/zalijepi plagijat podrazumijeva samo tehničko direktno prepisivanje to jest kopiranje i lijepljenje odlomaka ili čitavih poglavlja bez izmjena postignutih prepravljanjem. Kao što je navedeno uz preuzeto djelo se samo dodaje ime autora (plagijatora) koji je preuzeo originalni sadržaj.

2. ***Prerušeni plagijat (Disguised Plagiarism):*** govori se o prikrivenom plagijatu gdje se većina sadržaja preuzima s manjim izmjenama kako bi se prikrili preuzeti sadržaji. Ovakva vrsta plagijata u velikoj mjeri podrazumijeva korištenje prepravljanja gdje se dodatno umeću riječi ili se neki oblici ili dijelovi fraza namjerno brišu. Prepravljanje si uzima toliko maha da se mijenjaju i glagolski oblici i vremena. Poduzimaju se i vrlo česti pokušaji parafraziranja bez navođenja autora djela.
3. ***Plagijat prevođenjem (Plagiarism by Translation):*** kad je tekst preuzet iz jednog jezika ili govornog područja, prevođenje se može provoditi ručno ili uz pomoć pomoć automatskog prevođenja, bez navođenja izvora.Za primijetiti je da kod ovakvog plagiranja, iako se radi o plagiranju kojeg otkrivaju softveri za detekciju, treba jako dobro baratati jezicima i značenjima termina. Ovo je iznimno zahtjevna razina detektiranja plagijata jer kao prvo mora postojati obrazac poznavanja jezika gdje nije dovoljan samo prevoditelj izvan odgovarajućeg konteksta nego kontekstualni prevoditelj koji ima mogućnost uviđanja u prijenos značenja koje je preneseno prijevodom pojmova.
4. ***Protresi/zalijepi (Shake & Paste):*** predstavlja varijaciju kopiraj/zalijepi plagijata gdje su sastavi i odlomci organizirani bez nekog logičnog reda i ne postoji jasan prijelaz između odjeljaka, odlomaka i cjelina. Radi se o plagijatima koji podrazumijevaju istovremeno kompilaciju sadržaja više različitih autora. Pritom dolazi do miješanja različitih stilova pisanja koji dovode do nepostojanja preglednog i jasnog konzistentnog prijelaza između odjeljaka. Takav plagijat je vrlo često zbrkan i nejasan jer se prožimaju različiti stilovi pisanja koji nisu svojstven jednoj osobi.
5. ***Strukturni plagijat (Structural plagiarism):*** vrsta plagijata gdje se osim konkretnog sadržaja bez navoda i referenci koriste i citati drugih osoba naravno bez navođenja i referenciranja. Ova vrsta plagijata je vrlo zahtjevna za detekciju. Problem ovog plagijata je taj što se primjerice u jednom odlomku koriste osim izvornih riječi nekog autora i riječi autora kojeg je prethodni autor citirao. Dakle citati unutar originalnog djela koje se kopira u svrhu plagiranja, navode se kao vlastiti sadržaj. S time da se ni na koji način ne označava (referencom ili parafrazom) da se radi o nekom citatu kojeg je autor kojeg plagiramo koristio. Zbog toga je ovakav plagijat vrlo teško otkriti jer je i za softvere vrlo zahtjevan posao tražiti istovremeno više različitih izvora koji su kombinirani u plagijatu.

Pojavni oblici plagijata su samo tehnička komponenta preko koje se plagijat može jasno identificirati radi praćenja i prepoznavanja. Srž plagijata, nalazi se u otkrivanju uzroka čina plagiranja neovisno o pojavnom obliku.

2. Računalna detekcija plagijata

Iz navedenog vidljivo je koliko je način plagiranja kompleksan te koliko vrsta ili inačica plagiranja postoji. Pretpostavka je da bi softveri trebali moći otkrivati sve te varijante i pojavne oblike plagijata. Gotovo svi softveri za realizaciju detektiranja plagijata, ostvareni su metodama analize tekstova na temelju sličnosti znakova. To znači da većina softvera kao fizički uspoređuju riječ po riječ, rečenicu po rečenicu i odlomak po odlomak. No usporedba se ne provodi uvijek istim redoslijedom to jest istim nizom koraka. Ovisno o kompleksnosti sadržaja te brzini detekcije, postoje i razvijeni algoritmi koji pospješuju detekciju. Dakle posebna vrsta plagijata je ***Shake & Paste*** plagijat odnosno protresi i zalijepi. Takav plagijat je vrlo teško obraditi jer se sastoji od gotovo nasumično uzetih i međusobno povezanih uzoraka (odlomaka ili dijelova rečenica). U poglavlju su prikazane najčešće korištene metode i algoritmi koji se koriste u svrhu uspoređivanja tekstova i detektiranja sličnosti. Naravno od mnoštva algoritama i poznatih metoda, uzete su najpoznatije odnosno najkvalitetnije. Detaljnim prikazom njihovog funkcioniranja te međusobnom usporedbom, dati će se pregled i sugestije u rješavanju problema detekcije plagijata. Računalni softver je samo alat poput kalkulatora za rješavanje problema no logika funkcioniranja odnosno algoritam njegovog rada, najbitnija je komponenta softvera.

2.1. Metode analize tekstova na temelju sličnosti znakovnih nizova

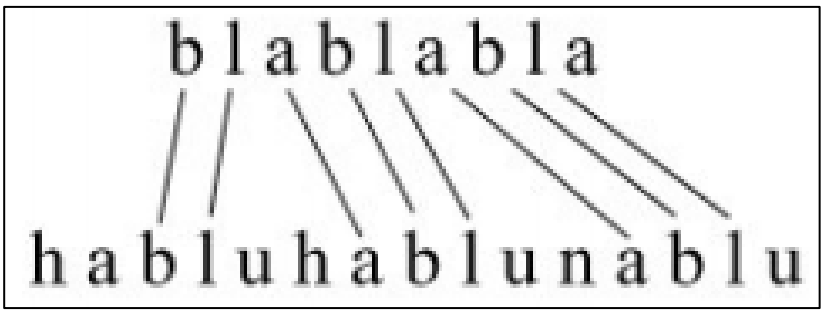
Analiza tekstova na temelju sličnosti nizova znakova podrazumijeva detaljan prolazak kroz svaku rečenicu sadržaja te uspoređivanje uzorka sa zadanim sadržajem kojem se utvrđuje razina plagiranja. Analiza teksta podrazumijeva korištenje cijelih uzoraka ili dijelova tih uzoraka. Na taj način postiže se istovremeno i brzina detektiranja, ali i dubina utvrđivanja sličnosti.

2.1.1. Najdulji zajednički podslijed

Metoda najduljeg zajedničkog podslijeda podrazumijeva usporedbu znakova odnosno znakovnih nizova između dva paralelna teksta. Izrazito se naglašava da „...treba razlikovati ovu metodu od metode nalaženja najdužeg zajedničkog podniza...“[1]. Podslijed podrazumijeva prije svega praćenje duljine nekog slijeda da bi se na temelju toga mogla napraviti usporedba sa slijedom paralelnog teksta koji se uspoređuje s tekstom u obradi. Dakle inicijalni smisao ove metode sastoji se u „...traženju znakovnog niza unutar drugog i kao rezultat vraća duljinu podslijeda...“[2]. Ono što se ovom metodom traži je uzorak identificiran u tekstu. Tekst može biti jedna rečenica unutar koje se pokušava identificirati jedna riječ da bi se na temelju toga mogla utvrditi sličnost s uspoređivanim tekstom.

Kao što je napomenuto u poglavljima s pojavnim oblicima plagijata, plagijati mogu nastati na temelju dodavanja ili ispuštanja pojedinih konstrukata riječi ili rečenica da bi se prikrila izvorna originalna verzija dokumenta. Ukoliko odaberemo kao uzorak jednu riječ, na temelju nje možemo utvrditi cijelu modificiranu rečenicu dodatnim (suvišnim) riječima. Ako imamo rečenicu modificiranu dodatnim umetnutim riječima, odabirom jednog od uzoraka znakova (riječi) se pokušamo usidriti u toj rečenici. Dakle između odabranih uzoraka može biti „...proizvoljan broj umetnutih znakova...“[3]. Primjerice ako odaberemo dva znakovna niza: „blablabla“ i „habluhablunablu“. Ta dva niza moramo napisati jedan iznad drugoga da bi bilo lakše utvrditi sličnosti.

[1] Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda, Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2015.)
[2] Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2015.)
[3] Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2015.)

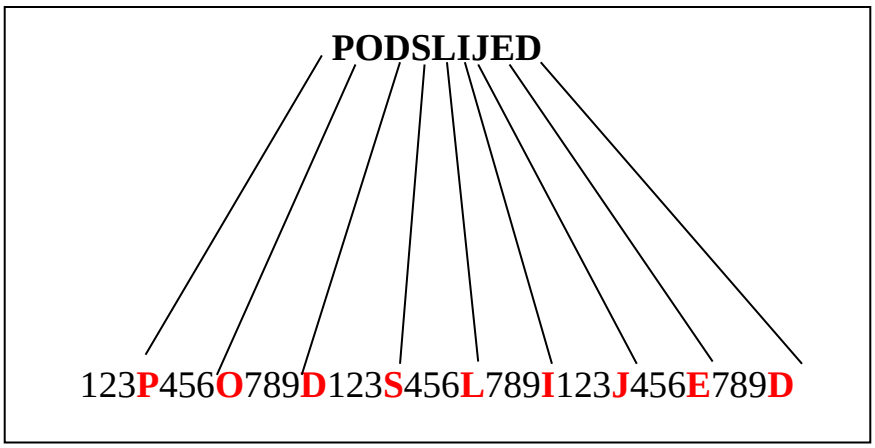


Slika 1: Uspoređivanje uzorka u podslijedu[4]

Prvi slijed znakova „blablabla“ predstavlja uzorak na temelju kojeg će biti provedena usporedba s drugim slijedom znakova „habluhablunablu“. Slijed znakova „blablabla“ nazivamo ujedno i podslijedom. Svaki uzorak je na neki način podslijed jer slova koja se pojavljuju u njemu, sadržana su u slijedu znakova kao u primjeru „habluhablunablu“. Dakle ako u nastavku okomitim linijama povezujemo slova koja su zajednička u oba slijeda znakova (u uzorku i originalnom slijedu znakova) dobivamo da su slova iz uzorka (prvog slijeda znakova) ekvivalentna sa slovima drugog slijeda znakova koja također odabiremo po redu. Ta uparena slova nazivamo podslijedom. Naravno kod ove metode je važno da se znakovi pronalaze istim redoslijedom kako su bili definirani, a ne nekim obrnutim redoslijedom, jer bi se na taj način izgubila konzistentnost prvotno izabranog i definiranog uzorka.

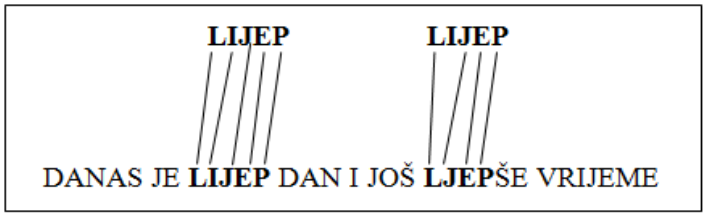
Primjerice na slici 1 uzorak „blablabla“ započinje tek s prvim slovom „b“ u originalnom slijedu, iako je moguće prije slova „b“ prepoznati i znak „a“ koji se također nalazi definiran u uzorku. Prateći uvijek isti redoslijed definiranog uzorka, konzistentno se identificira uzorak u originalnom slijedu znakova. Primjerice kao uzorak možemo uzeti riječ „podslijed“, a kao originalni slijed znakova možemo dodati proizvoljan broj znakova koji će biti u ovom primjeru biti dopunjen brojevima.

[4] Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2015.)



Slika 2: Traženje podslijeda

Pronađen je podslijed neovisno o tome što je originalni slijed znakova dopunjen brojkama. Kada u dva slijeda znakova oba započinju istim slovom, najsigurnije je spojiti ta dva slova kao dio podslijeda. Ukoliko se prvo slovo kao na slici 1 nalazi na nekom desnijem mjestu (primjerice na desnijem mjestu je bilo slovo „b“ kao prvo slovo uzorka) linija kojom se spajaju ta dva slova u oba niza se može preusmjeriti u lijevo bez da se uzrokuje presijecanje linija. To je najsigurniji način utvrđivanja da se radi o prvom slovu. U nekim se situacijama mogu prva slova razlikovati (podrazumijeva se nekoliko prvih slova), nemoguće je da će oba biti dio podslijeda, nego će biti potrebno bar jedno od njih (ili oba) ukloniti. Konačan slikovit primjer upotrebe ove metode može se provjeriti primjerom traženja i usporedbe uzorka „lijep“ u rečenici „Danas je lijep dan i još ljepše vrijeme.“ Ovaj primjer je istovjetan primjeru na slici 2. Traženje podslijeda, jer se uzorak „lijep“ nadopunjuje dodatnim znakovima „Danas je...dan i još ljepše vrijeme.“



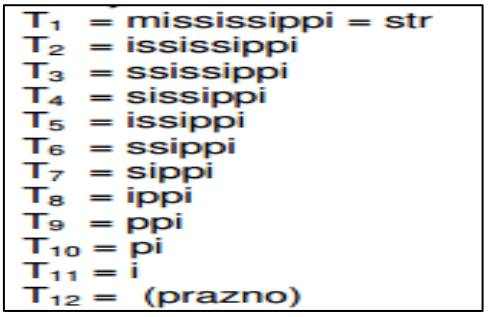
Slika 3: Traženje podslijeda u konkretnoj rečenici

Uzorak je identificiran na dva mjesta iako u riječi „ljepše“ nije identificiran cjelovit uzorak, ali su identificirana prva slova uzorka „lijep“. Podslijed i podniz su međusobno povezani iako su u određenim aspektima različiti, ali jedno bez drugoga bi u softverima za detekciju plagijata izazvalo brojne propuste, što opet rezultira nedostatnom identifikacijom plagijata.

2.1.2.Najdulji zajednički podniz

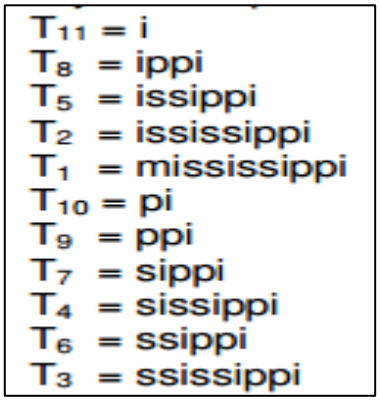
Podniz i podslijed su intuitivno vrlo srodni pojmovi ali kako je već naglašeno „...treba razlikovati ovu metodu od metode nalaženja najdužeg zajedničkog podniza...“[5]. To znači da se ova metoda razlikuje od metode najdulji zajednički podslijed po tome što se pod pojmom podniz smatra skup znakova iz originalnog znakovnog niza koji je povezan, dok se podslijedom smatra bilo koji podskup znakova iz originalnog niza koji se može dobiti brisanjem nula ili više znakova originalnog niza. Prema tome podniz je definiran kao neprekinuti niz originalnih znakova koji se u istom redoslijedu mora pronaći u drugom tekstu ili nizu dok se kod podslijeda u originalni niz znakova može dodati ili izbrisati proizvoljan broj znakova. Na primjer imamo originalni niz „banana“. Odbacivši sufikse i prefikse niza „banana“, podniz bi izgledao ovako „ana“. Za isti niz „banana“ odbacivanjem jednog, nijednog ili više ne nužno uzastopnih znakova, podslijed bi glasio „baaa“. Važno je za naglasiti da se u svrhu ove metode koristi takozvano sufiksno stablo. Prema tome stablo sufiksa je „...struktura podataka koja omogućava rješavanje raznih problema vezanih uz znakovne nizove u linearnom vremenu. Ako znakovni niz označimo sa str = t1 t2 t3...tn onda je Ti = ti...tn sufiks od str koji počinje na poziciji i...“[6].

[5] Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2015.)
[6] L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.)



Slika 4: Početno raščlanjivanje riječi mississippi na podnizove[7]

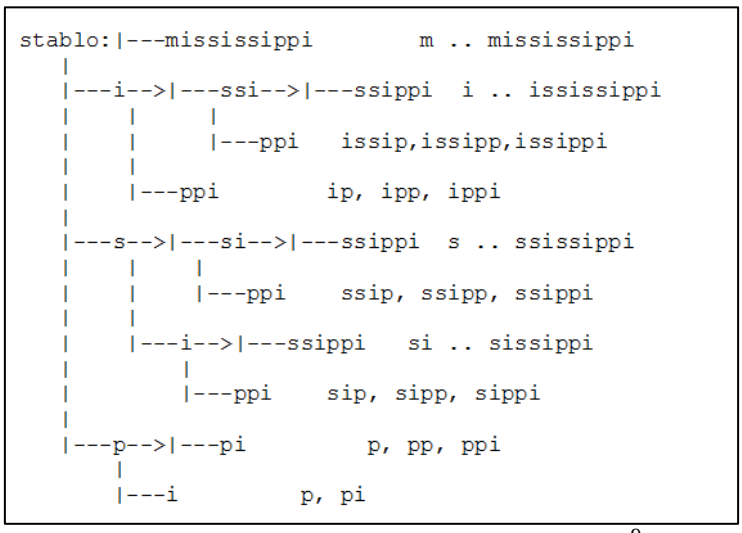
Za primjer možemo uzeti riječ mississippi. Metoda će raditi na način da se za zadanu riječ izdvajaju svi mogući podnizovi te iste riječi koje se kasnije sortiraju prema početnom slovu. Na slici 4 je prikazano početno raščlanjivanje riječi mississippi po parametrima T1-T12. Nakon raščlanjivanja, provodi se sortiranje po početnom slovu kojeg ovdje nazivamo sufiksom. Za primijetiti je da ukoliko sortiramo prefikse po slovima, neki od njih imaju zajedničke prefikse (npr. i,p,s).



Slika 5: Sortiranje prefiksa po slovima[8]

Sljedeći korak je izrada stabla sufiksa na način da se sufiksi sa zajedničkim prefiksom interpretiraju kao korijen u stablu. Točnije sufiksi sa zajedničkim prefiksom imaju zajednički korijen u stablu.

[7] L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.)
[8] L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.).

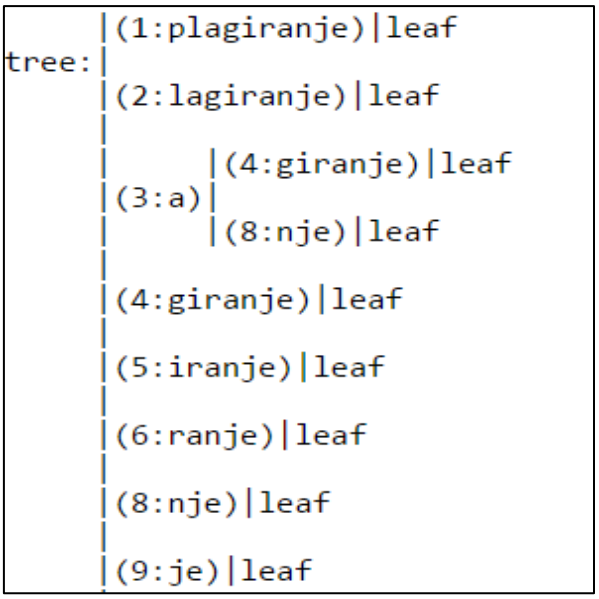


Slika 6: Sufiksno stablo riječi mississippi[9]

Najduži zajednički podniz se traži pomoću generaliziranog stabla sufiksa. Generalizirano stablo sufiksa sadrži sve sufikse više znakovnih nizova (izvornih tekstova programa) koje uspoređujemo. „...Čvorovi takvog stabla moraju biti označeni ovisno da li pripadaju prvom, drugom ili oba osnovna niza...“[10]. Prema tome u slučaju riječi „mississippi“ zajednički korijeni su nad prefiksima **i,p,s**. Primjena ove metode je moguća i u određivanju sličnosti izvornih tekstova ako izgradimo općenito sufiksno stablo koje će sadržavati sve sufikse obaju tekstova. Kako bi usporedba bila moguća treba voditi računa da se za svaki čvor stabla zna kojem tekstu pripada: prvom, drugom ili oba teksta.

[9] L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.)
[10] L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.).

Ukoliko raščlanimo niz znakova „plagiranje“, dobit ćemo situaciju kao na slici 7. sufiksno stablo riječi „plagiranje“.



Slika 7: Sufiksno stablo riječi „plagiranje“

Za primijetiti je da riječ plagiranje ima samo dvije grane od kojih je najdublji čvor (9:**je**). Može se zaključiti da je najdulji zajednički podniz „je“ prema kojem će se vršiti daljnje pretraživanje u uspoređivanju nizova znakova. Identifikacija plagijata pomoću softvera ne bi bila potpuna da ne postoje metode koje „mjere“ udaljenost u nizovima i podsljedovima.

2.1.3.Hammingova udaljenost

Hammingova udaljenost kao metoda za utvrđivanje plagijata se upotrebljava u teorijama informacija gdje je udaljenost dvaju nizova znakova iste duljine definirane kao broj bitova ili lokacija na kojima ta dva postojeća niza ne sadrže identične znakove. Dakle otkriva se plagiranje na temelju prepoznavanja mijenjanja ili modificiranja originalnih znakova. To dokazuje definicija Hammingove udaljenosti prema (Nikolić, 1987) kao „...Hammingova udaljenost (HD) između dvije riječi je broj bitova u kojima se one razlikuju...“ U nastavku plagijat se utvrđuje na temelju pogrešaka u bitovima odnosno lokacijama. Dakle „...Ako dvije riječi imaju Hammingovu udaljenost x, tada je potrebno pogriješiti x bitova da bi se jedna kodna riječ pretvorila u drugu...“ (ibid).

Za primjer možemo uzeti riječ koja je pretvorena u bitove. Ovo su samo simbolični nazivi riječi koje su pretvorene u binarni kod.

Primjer 1(Riječi u binarnom kodu):

riječ 1: 01100110
riječ 2: 00101001

Plavom bojom su prikazani originalni bitovi riječi 1, a crvenom su označene „greške“ odnosno promijenjene lokacije bitova. Dakle Hammingova udaljenost između riječi 1 i riječi 2 je pet bitova ili pet pogrešaka lokacije. Metoda Hammingove udaljenosti doslovno, mjeri minimalni broj supstitucija ili zamjena koje su potrebne za promjenu jedne riječi u drugu riječ ili broj grešaka prilikom transformacije jedne riječi u drugu. Osnovni uvjet Hammingove udaljenosti je da nizovi znakova moraju imati nužno isti broj elemenata što naposljetku implicira da se dogodila samo zamjena, a ne i brisanje ili dodavanje kao kod drugih metoda.

Primjer 2 (Hammingova udaljenost riječi „Bijeg“ i „Tijek“):

Bijeg

Tijek

Hammingova udaljenost između ove dvije riječi je jednaka dvije lokacije ili dvije greške odnosno dva bita. Promijenjene su pozicije riječi **B** u **T** i **g** u **k**. Ovo je bio prikaz jedne mogućnosti primjene mjerenja udaljenosti, no postoji i još nekoliko oblika bržeg i efikasnijeg načina mjerenja udaljenosti kao što je Levensteinova udaljenost.

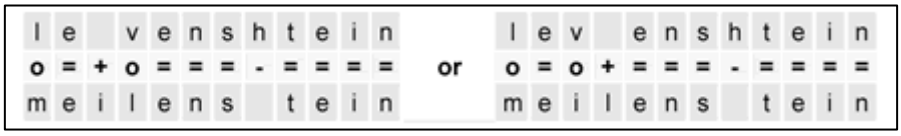
2.1.4.Levensteinova udaljenost

Levensteinova udaljenost „...izračunava najmanji broj operacija koje su potrebne za transformaciju jedne riječi u drugu odnosno transformaciju jednog znakovnog niza u drugi...“[11] To bi u nastavku značilo da se Levensteinova udaljenost definira kao broj potrebnih akcija s kojima se jedan znakovni niz pretvara u drugi pri čemu se zadovoljava mogućnost plagijatora da mijenja, dodaje ili briše znakove. Rabi se matrica izračunavanja akcija tipa (m,n) gdje se zamjenjuje m-prefiks s n-prefiksom.

		m	e	i	l	e	n	s	t	e	i	n
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4

Slika 8: Levensteinova matrica izračunavanja udaljenosti[12]

Za primijetiti je da se s lijeve strane riječ „Levenshtein“ nalazi zapisana u stupcu s n-prefiksom, a riječ „meilenstein“ se nalazi s m-prefiksom u retku matrice. Matrica može biti popunjena s gornje lijeve strane na donjem desnom kutu. Svaka vodoravna ili okomita aktivnost transformacije odgovara akciji brisanja ili dodavanja. Svaka promjena znaka označava se brojem transformacija u obliku brisanja ili dodavanja. Normalno postavljena na 1 za svaku od operacije. U ovom slučaju matrice imamo dijagonalnu promjenu što znači da je baš svaki znak početne riječi „Levenshtein“ bio podvrgnut transformacijama.



Slika 9: Mogući putevi kroz matricu[13]

[11] The Levenshtein Algorithm (2012) Dostupno na http://www.levenshtein.net/ (preuzeto 23.04.2015.)
[12] The Levenshtein Algorithm (2012) Dostupno na http://www.levenshtein.net/ (preuzeto 23.04.2015.)
[13] The Levenshtein Algorithm (2012) Dostupno na http://www.levenshtein.net/ (preuzeto 23.04.2015.)

Putovi zamjene riječi „Levenshtein“ mogu se svesti na riječ „meilenstein“ ili riječ „meilenstein“. Sve ovisi o potrebi i interesu plagijatora.

Primjer 3 (Levensteinova udaljenost znakova „lijek“ i „riječi“):

lijek -> rijek

rijek->riječ

riječ->riječi

Levensteinova udaljenost znakova „lijek“ i „riječi“ iznosi tri. Dakle napravljene su točno tri promjene da bi se lijek pretvorio u riječi. Prva promjena je podrazumijevala zamjenu početnih slova „l“ slovom „r“ . Drugi korak zamjene je podrazumijevao zamjenu posljednjeg znaka „k“ u znak „č“. I u trećem koraku je nad znakom „riječ“ dodan znak „i“, čime je znak „lijek“ potpuno transformiran u znak „riječi“. Iako je vrlo brza i efikasna kao metoda detektiranja plagijata, utvrđeni su poneki propusti u radu. Stoga je ova metoda modificirana do razine Damerau-Levensteinove udaljenosti.

2.1.5.Damerau-Levensteinova udaljenost

Damerau-Levensteinova udaljenost jest proširenje postojeće Levensteinove udaljenosti. Dakle definicija Levensteinove udaljenosti je glasila: Levensteinova udaljenost „...izračunava najmanji broj operacija koje su potrebne za transformaciju jedne riječi u drugu odnosno transformaciju jednog znakovnog niza u drugi...“[14]. Damierau-Levensteinova udaljenost potom podrazumijeva „...dodavanje akcije kojom se lokacije dva susjedna znaka jednog niza mogu zamijeniti...“ (Damerau, 1964). S time se može iznijeti pretpostavka da su riječi nekog teksta kratke te da je broj pogrešaka prilikom pisanja rijetko veći od dvije lokacije ili pogreške.

[14] The Levenshtein Algorithm (2012) Dostupno na http://www.levenshtein.net/ (preuzeto 23.04.2015.)

Primjer 4 (Damerau-Levensteinova udaljenost):

tuorka->utorka

utorka->utorak

Za primijetiti je da se zamjena susjednih znakova dogodila na prefiksima riječi „tuorka“ i „utorka“ u jednom slučaju. To se broji kao zamjena jedne lokacije ili identifikacija jedne greške. Druga zamjena se odnosi na riječi „utorka“ i „utorak“ gdje se zamjena susjednih znakova dogodila na kraju riječi to jest na sufiksima. Tu se identificirala isto jedna lokacija ili greška. Stoga valja zaključiti da se rade maksimalno dvije pogreške prilikom pisanja. No ne treba zanemariti mogućnost da se može pojaviti i neka druga varijanta pogreške ili zamjene što bi značilo da je početna pretpostavka Damerau-Levensteinove udaljenosti zamjene lokacija dva susjedna znaka relativna. Metode su se pokazale vrlo uspješnima u radu no važno je sagledati i upoznati funkcionalnost algoritama koji pospješuju metode analize tekstova na temelju sličnosti znakova.

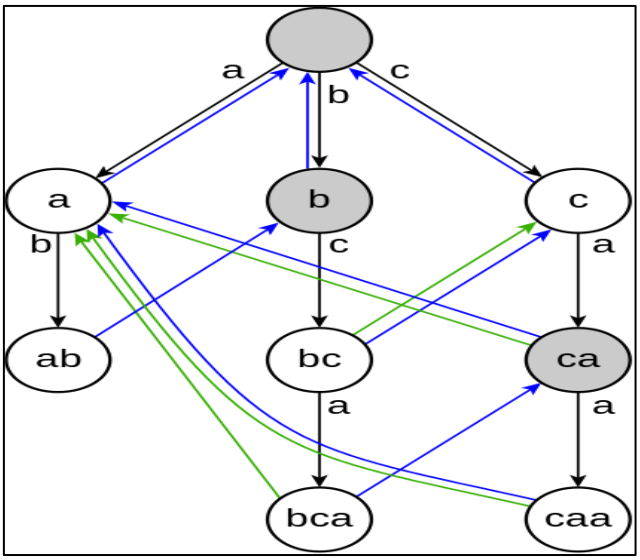
2.2.Algoritmi detekcije plagijata

U svrhu detekcije plagijata u izvornih tekstovima, razvijen je veliki broj algoritama za brzo pretraživanje nizova znakova na temelju uzoraka. Uvjet za odabir algoritma za pretraživanje nizova znakova je prvenstveno mogućnost detektiranja više uzoraka odjednom.

2.2.1.Aho-Corasick algoritam

Aho-Corasick algoritam je „...klasično i skalabilno rješenje za određivanje točnog podudaranja znakova te naširoko poznat algoritam za pronalaženje više uzoraka...“ (Vidanagamachchi i sur, 2012). Implementiran je od strane Alfreda V. Ahoa i Margaret J. Corasick. Pretpostavka za rad ovog algoritma je poznato stablo sufiksa korišteno u metodama analize tekstova na temelju sličnosti znakovnih nizova. Primjer ovakvog algoritma možemo prikazati pomoću jednostavnog uzorka (bca).

Primjer 5 (Primjena Aho-Corasick algoritma s [bca] uzorkom)



Slika 10: Aho-Corasick algoritam traženja otisaka izvornih tekstova[15]

[15] Algoritm of the Week: Aho-Corasick String Matching Algorithm (2013) Dostupno na http://architects.dzone.com/articles/algorithm-week-aho-corasick (preuzeto 23.04.2015.)

Dakle struktura podataka ima po jedan čvor za svaki prefiks svakog znaka. Primjerice ako primijenimo naš uzorak (bca), kreirat će se čvorovi (bca), (bc), (b) i (). Možemo zamijetiti da se sufiksno stablo sastoji od crne usmjerene grane i plave usmjerene grane.

Samim time postoji crna grana od (bc) do (bca) uzorka (bca). Zatim postoji plava usmjerena sufiks grana od svakog čvora do čvora koji je najdulji mogući striktni sufiks tog čvora u grafu. U ovom primjeru za čvor (caa) striktni sufiksi su (aa), (a) i (). Najdužim striktnim sufiksom u grafu od (aa), (a) i () je (a), tako da postoji plava grana od (ca) do (a). Isto tako postoji i zelena grana sufiksa od svakog čvora do sljedećeg čvora do kojeg se može doći prateći plave grane. Primjerice, postoji zelena grana od (bca) do (a) zbog toga što je (a) prvi čvor do kojeg se dolazi putem plave grane do (ca), a onda do (a). Na svakom koraku, trenutni čvor se produljuje tražeći vlastito dijete, a ako ono ne postoji, onda nalazi sufiks. Kada algoritam dođe do čvora, izbacuje sve moguće ulaze koji završavaju na trenutnoj poziciji karaktera u unesenom tekstu. Ovaj algoritam je takoreći dobar. No kvalitetu algoritma moguće je utvrditi tek nakon usporedbe njegove funkcionalnosti s drugim algoritmima.

2.2.2.Boyer-Moore algoritam

Boyer-Moore algoritam „...uspoređuje simbol uzorka i simboli teksta s desna na lijevo, počevši od zadnjeg simbola uzorak...“ (Melichar, 2006). Razvijen je od strane Roberta S. Boyera i J. Stroter Moorea. Princip rada ovog algoritma temelji se na poravnavanju uzoraka sa zadanim tekstom. To znači da se uzorak doslovno pomiče unutar teksta da bi se ostvarilo podudaranje. Boyer-Moore algoritam koristi informacije dobivene iz pretprocesiranja da preskoči što više poravnanja, kako bi se postigla odgovarajuća brzina prolaženja kroz tekst kroz utvrđena podudaranja.

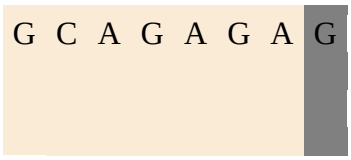
Primjer 6 (Primjena Boyer-Moore algoritma)[16]

U prvom koraku imamo niz znakova nekog teksta

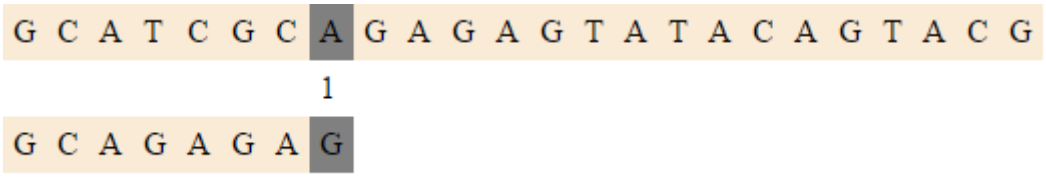
G C A T C G C A G A G A G T A T A C A G T A C G

[16] Boyer-Moore Algorithm MET (1997) Dotstupno na http://www-igm.univ-mlv.fr/~lecroq/string/examples/exp14.html (preuzeto 23.04.2015.)

i uzorak

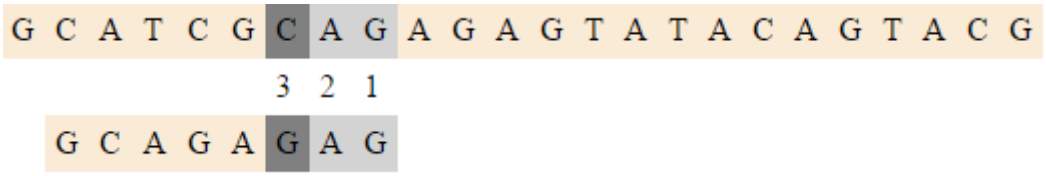


Prvi pokušaj



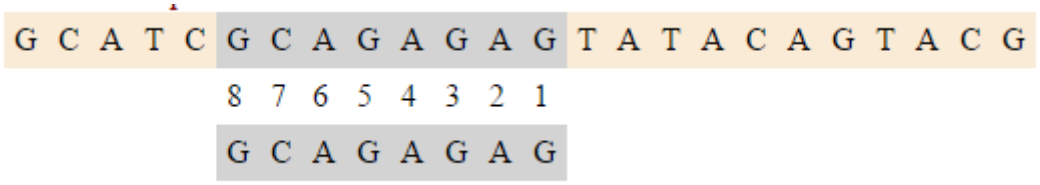
Budući da funkcioniranje algoritma zahtjeva kretanje od lijeva prema desnom kraju, prvotni položaj uzorka mora biti smješten skroz lijevo. Za početak nema podudaranja.

Drugi pokušaj



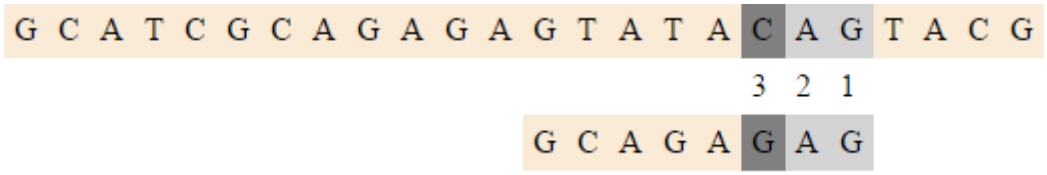
Pomaknuli smo se za jedan korak dalje i dogodilo se prvo podudaranje slova A. Algoritam je nastavio dalje pomicati uzorak prema desnom kraju i dolazi do podudarnosti slova A i G. Sveukupno su napravljena tri koraka pomicanja u desnu stranu.

Treći pokušaj



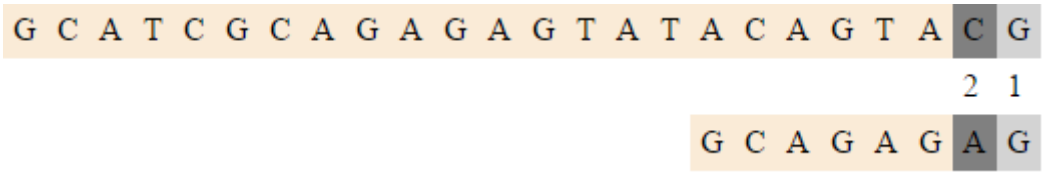
U trećem pokušaju, pomicali smo uzorak sveukupno osam koraka u desnu stranu i dobili smo prvo cjelovito podudaranje. Naravno postupak još nije gotov jer algoritam identificira još jednu podudarnost. Budući da je ova podudarnost ostala zapamćena, algoritam se više neće vraćati na nju, nego će je preskočiti.

Četvrti pokušaj



U četvrtom pokušaju, pronađena su dva nova podudaranja slova A i G. Algoritam se pomiče za tri mjesta u desnu stranu. No podudarnost cjelovitog uzorka nije postignuta. Algoritam bilježi djelomičnu podudarnost uzorka s nizom znakova teksta kako se više ne bi vraćao na njega.

Peti pokušaj



Peti i ujedno posljednji pokušaj podrazumijeva podudaranje slova G u uzorku što znači da je uzorak pomaknut za dodatna dva mjesta u desno. Cjelovito podudaranje nije postignuto, ali su identificirana mjesta u tekstu gdje je prepoznat dio zadanog uzorka. U odnosu na prethodni algoritam, Boyer-Moore algoritam je prosječno brz u radu, ali i praktično koristan pogotovo ako se upotrebljava u metodama s podslijedom ili podnizom. Naravno praktičnost nije upitna, već je upitna brzina rada u usporedbi sa sljedećim Rabin-Karp algoritmom.

2.2.3.Rabin-Karp algoritam

Algoritam Rabin-Karp su razvili Michael O. Rabin i Richard M. Karp 1987. Godine. Algoritam omogućava pretraživanje znakovnih nizova „...koristeći funkciju raspršenja...“ (Cho i sur, 2004). Algoritam je poznat po svojoj iznimnoj brzini rada. Brzina Rabin-Karp algoritma temelji se na brzom uspoređivanju znakovnih nizova. Osnovna ideja je korištenje sažetaka k-grama. Zbog toga je glavna ideja usporediti sve sažetke k-grama. Pod k-gramima se podrazumijevaju parovi znakova koji se međusobno uspoređuju. To se radi na način da se izračunaju sve te vrijednosti za duge znakovne nizove unutar kojih se traži neki podniz. Prema (ibid.), deklaracija algoritma je sljedeća:

Neka je k-gram k-znamenkasti broj c1...c u nekoj bazi ***b***. Kao funkcija računanja sažetka se uzima:

$$H(\mathrm{C}_1 \dots C_k) = C_1 \times b^{k-1} + C_2 \times b^{k-2} + C_2 \times b^{k-3} + \cdots + C_k$$

Kako bi izračunali vrijednost sažetka novog k-grama potrebno je postaviti sljedeći račun:

$$H(C_2 \dots C_{k+1}) = (H(C_1 \dots C_k) - C_1 \times b^{k-1}) \times b + C_{k+1}$$

S obzirom da je b $^{k-1}$ konstanta svaka iduća vrijednost se računa pomoću dvije operacije zbrajanja, i dvije operacije množenja. Te operacije se uzimaju kao „modulo“ neke vrijednosti, najčešće najveća vrijednost za cijeli broj. Da bi se pronašao željeni uzorak unutar nekog znakovnog niza, Rabin-Karp algoritam putem deklarirane relacije izračunava vrijednost trenutnog detektiranog uzorka koja se uspoređuje s vrijednosti početnog zadanog uzorka. Ukoliko se vrijednosti poklapaju, radi se dodatna provjera ASCII koda oba uzorka i ukoliko se poklapaju, pronađen je željeni uzorak. U suprotnom algoritam prolazi dalje znakovnim nizom i uspoređuje vrijednosti relacija. Algoritam završava s radom tek kada dođe do kraja znakovnog niza neovisno o tome da li je već pronašao i detektirao željeni uzorak.

Funkcionalnost Rabin-Karp algoritma možemo sagledati kroz konkretan primjer.

Primjer 7 (Primjena Rabin-Karp algoritma)

Za znakovni niz ćemo uzeti jedanaest ne uzastopno slijednih brojeva. To su brojevi: **31415926535.** Kao uzorak pretraživanja ćemo uzeti **broj 26.** Duljina znakovnog niza je **11** brojeva. Vodit ćemo izvornom notacijom formule $\boldsymbol{P \times mod\ q}$ pri čemu **P** predstavlja par brojeva koji ulaze u uzorak usporedbe, zatim **q** koji se interpretira kao duljina znakovnog niza. Konačna deklaracija početnih veličina izgleda ovako:

Niz = 31415926535

P = 26

q = 11

$\boldsymbol{P \times mod\ q \longrightarrow 26 \times mod\ 11 = 4}$

Svaki par brojeva u zadanom nizu će sustavno prolaziti kroz formulu $P \times mod\ q$ i rezultat će se uspoređivati s dobivenom vrijednosti 4 dobivenom kao rezultat relacije $26 \times mod\ 11 = 4$. Kao konačno rješenje će se uzeti broj koji se poklapa s rezultatom relacije $26 \times mod\ 11 = 4$.

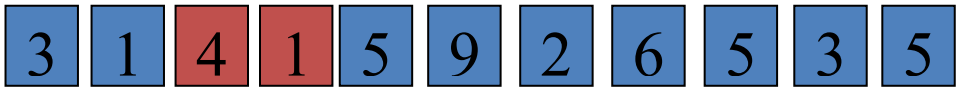
Princip rada algoritma kreće s lijeve strane znakovnog niza. U relaciju ulaze prva dva broja 3 i 1. Njihov rezultat će se usporediti s rezultatom početne relacije.

$31 \times mod 11 = 9 \neq 4$

Rezultat provedene relacije je 9 što se ne poklapa s vrijednosti 4. To znači da je potrebno prijeći na idući par brojeva. Sljedeći par brojeva je 1 i 4.

$14 \times mod 11 = 3 \neq 4$

Ni ovaj rezultat nije zadovoljavajući. Vrijednost relacije je 3. Provodi se daljnja usporedba idućih parova brojeva.



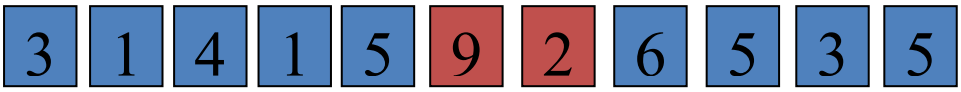
$41 \times mod 11 = 8 \neq 4$

$15 \times mod 11 = 4 = 4$

U ovom slučaju se dobilo poklapanje vrijednosti dobivene kao rezultat relacije 4. No uzorak 15 se ne poklapa s traženim uzorkom 26. Algoritam Rabin-Karp na temelju izračunate vrijednosti relacije pronalazi potencijalna mjesta u nizu znakova koja bi mogla odgovarati prvotnom zadanom uzorku. No to ne znači da će vrijednost kao rezultat relacije garantirati identičnost uzorka. Stoga Rabin-Karp algoritam mora napraviti dodatnu provjeru znakova. Točnije provjerava ASCII kod znakova niza. Primjerice ASCII kodovi za brojeve 1 i 5 su različiti od brojeva 2 i 6. Postupak rada algoritma se nastavlja na sljedećem paru brojeva.

$59 \times mod 11 = 4 = 4$

Dogodila se ista situacija kao u prethodnom koraku. Budući da nema poklapanja uzoraka, rad algoritma se provodi dalje.

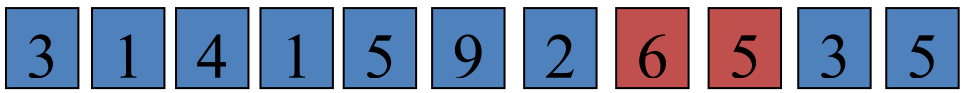


$92 \times mod 11 = 4 = 4$

Ponovno je došlo do poklapanja po rezultatu vrijednosti relacije, ali ne i prema usporedbi znakova putem ASCII koda.

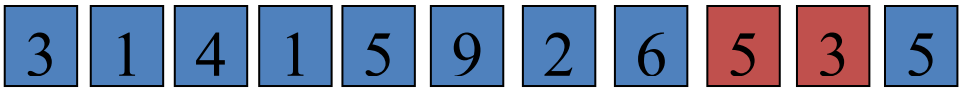
$26 \times mod 11 = 4 = 4$

Dobivene vrijednosti relacija trenutnog uzorka i početnog zadanog uzorka su se poklopile. Isto tako ASCII kod znakova 2 i 6 su identični i može se reći da se došlo do detekcije početnog uzorka. Iako je uzorak nađen i identificiran, algoritma nastavlja dalje s radom dok ne dođe do kraja znakovnog niza.



$65 \times mod 11 = 10 \neq 4$

Vrijednosti izračunatih relacije se nisu poklopile. Nema potrebe za daljnjom usporedbom ASCII koda znakova.



$53 \times mod 11 = 9 \neq 4$

$35 \times mod11 = 2 \neq 4$

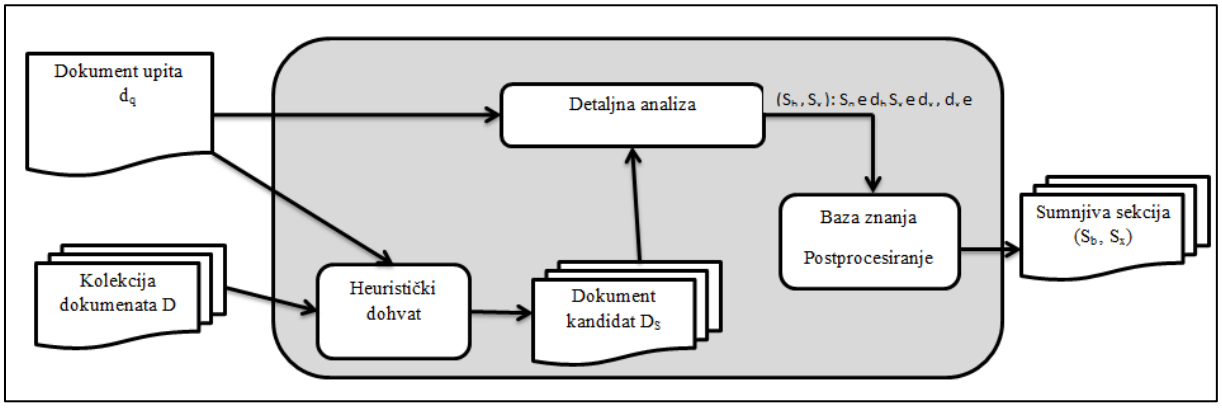
U posljednja tri koraka nije bilo nikakvog poklapanja. Algoritam je završio s radom kada je došao do kraja znakovnog niza. Početni uzorak zadan na početku je pronađen i identificiran prema broju izračunate vrijednosti relacije te dodatno provjeren putem usporedbe ASCII koda početnog uzorka i trenutnog uzorka detektiranog u znakovnom nizu. Algoritam je vrlo brz s obzirom na dvostruku provjeru koju radi prilikom detekcije uzoraka. U ovom poglavlju razjašnjeni su svi načini funkcioniranja sustava za detekciju plagijata preko metoda i algoritama funkcioniranja pojedinih metoda. Napravljen je pregled i zaokružena cjelina funkcioniranja jednog složenog mehanizma kao što je softver za detekciju plagijata. Sve dosadašnje metode pospješene algoritmima su počivale na detekciji plagijata nad konkretnim primjerom sadržaja dokumenta. No nigdje nije bila uključena usporedba sadržaja dokumenta sa zbirkom izvornih dokumenata. To je prepušteno metodama detekcije plagijata.

3. Metode detekcije plagijata

Kao što je već spomenuto u radu, plagijat se odnosi na uporabu tuđe informacije, jezika ili stila pisanja, bez prethodnog navođenja izvora. Stoga detekciju plagijata provodimo na temelju usporedbe dokumenata s izvornim dokumentima. Tu se nameće pitanje oko uspoređivanja konkretnih izraza, odlomaka i fraza između plagiranog i izvornog dokumenta, ali i pitanje oko detektiranja varijacija stila pisanja. Stoga prema (Potthast i sur, 2009) razlikujemo dvije metode detektiranja plagijata: vanjska detekcija plagijata i unutarnja detekcija plagijata.

3.1.Vanjska detekcija plagijata

Pod pojmom vanjske detekcije plagijata podrazumijevamo metodu usporedbe plagiranog dokumenta sa zbirkom izvornih dokumenata. „...Vanjska detekcija plagijata bavi se problemom pronalaženja preuzetih sadržaja u sumnjivim dokumentima na temelju referentnog korpusa...“ (Zeichner i sur, 2009). Taj referentni korpus podrazumijeva usporedbu s jednim ili više izvornih dokumenata što potvrđuju (ibid.) „...Vanjska detekcija plagijata je slična tekstualnom pretraživanju informacija...“. Sam radni model metode vanjske detekcije plagijata je predstavljen modelom „bijele kutije“.



Slika 11: Model „bijele kutije“ vanjske detekcije plagijata (Izvor: Alzahrani i sur, 2011)

Prema ovom modelu, detektiranje plagijata se provodi u tri koraka. „...Vanjska detekcija plagijata dijeli se u tri koraka: **heuristički dohvat**, **detaljna analiza**, a nakon analize, **post procesiranje temeljeno na znanju**...“ (Zeichner i sur, 2009). Korake koje su naveli (ibid.), mogu se interpretirati kao operacije, jer se u svakom od koraka izvodi odgovarajuća operacija.

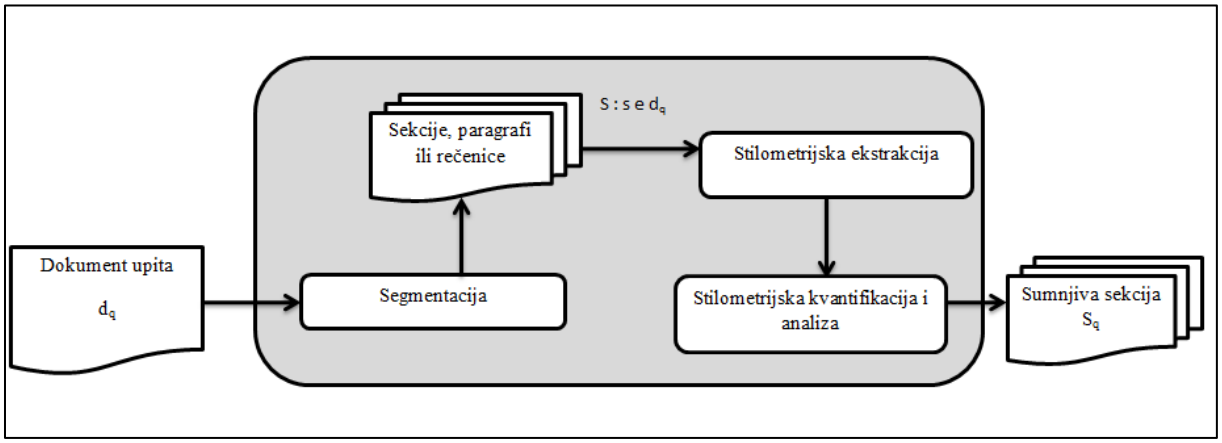
Na slici 11, na ulazu se nalaze veličine **dq** koja predstavlja sumnjivi dokument koji se uspoređuje s kolekcijom izvornih dokumenata, **D** gdje je **Dx** jedan izvorni dokument iz kolekcije **D**. Unutar bijele kutije, nalaze se tri operacije: heuristički dohvat, detaljna analiza, post procesiranje temeljeno na znanju. Na izlazu se kao rezultat dobiva specifičan odlomak ili fragment označen vrijednostima **Sq** odnosno jedan od pojavnih oblika plagijata i **Sx** odnosno plagirani fragment**.**

1. ***Heuristički dohvat*** povezan je s određenom skupinom dokumenata označenih kao-**Dx**, koji predstavljaju izvore plagiranja. **Dx** predstavlja preuzete dokumente iz skupine dokumenata **D**, nekom od metoda izvlačenja.
2. ***Detaljna analiza*** izvodi se uspoređujući **dq** sa **Dx**, odnosno sumnjivi (**dq**) dokument sa skupinom dokumenata koji su kandidati, odnosno potencijalni izvori plagiranja. „...To se izvodi usporedbom na razini neke jedinice npr. koristeći neku jedinicu usporedbe kao što je rečenica...“ (Potthast i sur, 2009)
3. ***Post procesiranje temeljeno na znanju*** izvodi se spajanjem malih otkrivenih jedinica, u odlomke.

Dakle, za konačni izlaz vrijedi da su male jedinice plagijata, odnosno fragmenti, (*sq*, *sx*), gdje je $sq \in dq$, $sx \in dx$, $dx \in Dx$, takvi da je *sq* uzorak plagijatima iz *sx* pri čemu vrijedi da *sq* predstavlja jedan od pojavnih oblika plagijata. Dakle vanjska detekcija plagijata, obavila je samo jedan dio posla u procesu detektiranja. Napravljena je usporedba sadržaja dokumenta s vanjskom zbirkom izvornih dokumenata. Da bi se detekcija upotpunila, potrebno je provjeriti i stil i jezik pisanja.

3.2. Unutarnja detekcija plagijata

Kao što je već spomenuto, metoda unutarnje detekcije plagijata, „...utvrđuje plagijat na temelju promjene stila u dokumentu...“ (Zeichner i sur, 2009). To bi značilo da se na temelju detektiranja stila pisanja identificira autor za kojeg je taj stil pisanja prepoznatljiv i specifičan. Dakle cilj metode unutarnje detekcije plagijata je stoga „...utvrditi potencijalni plagijat analizom dokumenta s obzirom na promjene napravljene u stilu pisanja. Provjera autorstva određuje je li tekst upitnog autorstva potječe od autora A, na način da se daju primjeri pisanja autora A. Za razliku od navedenog imenovanje autorstva ima za cilj pripisati neki dokument d, nepoznatog autorstva nekom od autora iz skupa autora D, koji je sačinjen od primjera pisanja te nekolicine autora...“ (Stein, Lipka, Prettenhofer, 2010). Da bi utvrdili funkcionalnost ove metode, potrebno je vizualizirati njezin rad pomoću bijele kutije.



Slika 12: Model „bijele kutije“ unutarnje detekcije plagijata (Izvor: Alzahrani i sur, 2011)

Kod ovog modela postoje samo dvije veličine ispitivanja: **dq** kao sumnjivi dokument, **D** kao referentna zbirka dokumenata. Unutar bijele kutije nalaze se tri koraka ili operacije: „...**segmentacija,, stilometrijsko izvlačenje ili ekstrakcija**, **stilometrijska kvantifikacija i analiza**...“ (Alzahrani i sur, 2011). Kao izlaz, dobiva se sumnjivi segment ili manja jedinica u obliku rečenice,, paragrafa ili sekcije.

1. ***Segmentacija*** sumnjivi dokument **dq** dijeli na manje dijelove ili segmente kao što su rečenice, paragrafi i sekcije.
2. ***Stilometrijsko izvlačenje ili ekstrakcija*** doslovno izvlači stilometrijske značajke iz različitih segmenata. Pod stilometrijskim značajkama podrazumijeva se da „...autor ima razvijeni stil pisanja, autor svijeno ili nesvjesno koristi obrasce za izgradnju rečenica i vlastiti vokabular riječi...“ (Eissen, Stein, Kulig, 2007)

3. ***Stilometrijska kvantifikacija i analiza*** kao operacija detektiranja plagijata, analizira varijacije različitih značajki stila pisanja. Pod stilometrijskim značajkama se podrazumijeva: „...1) statistiku teksta kroz različite leksičke značajke, na razini riječi i znakova, 2) sintaktičke značajke, na razini rečenica, 3) semantičke značajke koje se odnose na sinonime, funkcionalne riječi i semantičke ovisnosti, 4) posebne značajke vezane za organizaciju teksta, sadržajem određene ključne riječi i druge specifične značajke određene jezikom pisanja...“ (Stamatatos, 2010). Dakle, za konačni izlaz vrijedi da je fragment ili dio **Sq**, pri čemu je $sq \in d_q$ takav da **Sq** ima kvantificiranu značajku stila pisanja različitu od drugog fragmenta **S** u **dq**.

Metoda unutarnje i vanjske detekcije plagijata s osnovnim koracima analize, primjenjuju se u alatima za detekciju plagijata. Svi ti koraci i metode detekcije plagijata mogu se shvatiti i na globalnoj razini. Naime ovdje je napravljena detekcija na lokalnoj razini, koristeći samo sadržaje radova koji se nalaze na lokalnom repozitoriju. No alati za detekciju plagijata podrazumijevaju takozvanu online detekciju koja proširuje koncept detekcije plagijata na sadržaje s udaljenih izvora, a ne samo izvora koji su poznati lokalnom repozitoriju nekog sustava.

4. Literatura

1. L. Allison, članak Suffix trees, Dostupno na http://www.allisons.org/ll/AlgDS/Tree/Suffix/ (preuzeto 10.04.2015.)
2. Alzahrani M, Salim N, Ajith A (2011) Understanding Plagiarism Linguistic Patterns,Textual Features, and Detection Methods Dostupno na http://www.academia.edu/2724269/Understanding_Plagiarism_linguistic_patterns_textual_features_and_detection_methods (preuzeto 23.04.2017.
3. Beasley J, D (2006) The Impact of Technology on Plagiarism Preventionand Detection, Plagiarism: Prevention, Practice and Policies 2004 Conference yberspace, Computer Fraud and Security, Elsevier Science
4. Baždarić K, Pupovac V, Zulle L, Petrovečki M (2009) Plagiranje kao povreda znanstvene i akademske čestitosti, Dostupno na http://hrcak.srce.hr/38691 (preuzeto 10.04.2017.)
5. Cho C, Lee S, Tan C, Tan Y (2004) Network Forensics on Packet Fingerprints, Dostupno na http://link.springer.com/chapter/10.1007%2F0-387-33406-8_34#page-1 (preuzeto 10.04.2017.)
6. Eissen M, Stein B, Kulig M (2007) Plagiarism detection without reference collections, in Advances in Data Analysis
7. Hrannabuss S (2001) Contested texts: issues of plagiarism, Library Management MCB University Press
8. Joy M, Luck M (1999) Plagiarism in Programming Assignments, IEEE Transactions of Education
9. Melichar B (2006) Text searching algoritms Dostupno na http://www.stringology.org/athens/TextSearchingAlgorithms/tsa-lectures-2.pdf (preuzeto 10.04.2017.)
10. Nikolić B (1987) Jednostavna metoda za analizu promjena na jednom entitetu opisanom nad skupom kvalitativnih varijabli [http://hrcak.srce.hr/108291 preuzeto 02.04.2017.]
11. Olsson J (2010) Forenzička lingvistika, Zagreb

12. Potthast M, Stein B., Eiselt A, Barron-Cedeno A, Rosso P (2009) Overview of the 1st International Competition on Plagiarism Detection Dostupno na http://web.archive.org/web/20120402050919/http://www.uni-weimar.de/medien/webis/research/events/pan-09/pan09-papers-final/potthast09-overview-first-international-competition-plagiarism-detection.pdf (preuzeto 02.04.2017.)

13. Samuelson P (1994) Self-Plagiarism or Fair Use?, Communications of the ACM.

14. Stamatatos E (2010) A survey of modern authorship attribution methods

15. Stein B, Lipka N, Prettenhofer P (2010) Intrinsic plagiarism analysis, Language Resources & Evaluation

16. Vidanagamachchi S, M, Dewasurendra S, D, Ragel R, G, Niranjan M (2012) COMMENTZ-WALTER: ANY BETTER THAN AHOCORASICK FOR PEPTIDE IDENTIFICATION? Dostupno na http://www.ijorcs.org/uploads/archive/Vol2-Issue-06-04-commentz-walter-any-better-than-aho-corasick-for-peptide-identification.pdf (preuzeto 10.04.2017.)

17. Weber-Wulff D, Möller C, Touras J, Zincke E (2013) Plagiarism Detection Software Test 2013, Berlin Dostupno na http://plagiat.htw-berlin.de/software-en/test2013/report-2013/ (preuzeto 02.04.2017.)

18. Zeichenr M, Muhr M, Kern R, Granitzer M (2009) External and Intrinsic Plagiarism Detection Using Vector Space Models Dostupno na http://ceur-ws.org/Vol-502/paper9.pdf (preuzeto 10.04.2017.)

19. The Levenshtein Algorithm (2012) Dostupno na http://www.levenshtein.net/ (preuzeto 23.04.2017.)

20. Algoritm of the Week: Aho-Corasick String Matching Algorithm (2013) Dostupno na http://architects.dzone.com/articles/algorithm-week-aho-corasick(preuzeto 23.04.2017.)

21. Boyer-Moore Algorithm MET (1997) Dotstupno na http://www-igm.univ-mlv.fr/~lecroq/string/examples/exp14.html (preuzeto 23.04.2017.)

22. Design and Analysis of Algorithms, opis algoritama za pronalaženje najduljeg zajedničkog podslijeda Dostupno na http://www.ics.uci.edu/~eppstein/161/960229.html (preuzeto 10.04.2017.)

Zoran Hercigonja, rođen je 13.04.1990. godine u Varaždinu. Osnovnu školu "Petar Zrinski " završio je 2005. godine u Jalžabetu. Iste godine upisuje Elektrostrojarsku školu Varaždin, smjer "Tehničar za računalstvo" i uspješno položenom maturom završava 2009. godine. Daljnje školovanje nastavlja na Fakultetu organizacije i informatike Varaždin 2009. godine.Preddiplomski studij "Informacijski sustavi" završava 2013. godine. Nastavlja studirati na istom fakultetu na diplomskom studiju za informatičare usmjerenja "Informatika u obrazovanju" koji završava 2015. godine s velikom pohvalom (Magna cum laude). Akademske godine 2013/2014. dobitnik je dekanove nagrade za najboljeg studenta diplomskog studija informatike. Pripravnički staž obavlja na Drugoj gimnaziji Varaždin u razdoblju između 2015. i 2016. godine. Državno-stručni ispit za nastavnika informatike uspješno polaže na V. gimnaziji Zagreb 2017. godine. Radi kao profesor informatike i matematike. Autor je više stručnih radova koje redovito objavljuje u časopisima: Pogled kroz prozor, International Journal of Digital Technology and Economy, Matematika i škola, zborniku Carnetove korisničke konferencije te u Zborniku radova Veleučilišta u Šibeniku.

ISBN 978-953-59549-7-2