

فهرست مطالب

خلاصه	۵
مقدمه	۷
فصل اول : کلیات	۹
تحقیقات مرتبط و تاریخچه شناسایی سرقت ادبی	۱۰
سرقت ادبی چیست	۱۲
اهمیت سرقت ادبی	۱۳
مقیاس مشکل	۱۶
دسته‌بندی‌های سرقت ادبی	۱۷
تحقیقات انجام شده از نظر دسته بندی روش‌های شناسایی	۱۸
ردیاب ها	۱۸
شباهت رشته‌ای	۱۹
روش‌های مبتنی بر ساختار	۲۰
روش‌های دسته‌ای	۲۱
مدلهای نحوی	۲۱
روش‌های چند زبانی	۲۲
روش‌های معنایی (صرفی)	۲۲
فصل دوم : سرقت ادبی درون زبانی	۲۵
روش‌های مبتنی بر گرامر [۱]	۲۶
روش معنایی	۲۷
روش ترکیبی گرامر-معنایی	۲۷
نمونه‌هایی از روش‌های تشخیص سرقت ادبی درون زبانی	۳۱
تشخیص سرقت ادبی بر مبنای اطلاعات ساختاری [۳]	۳۱
نمایش متن	۳۱

۳۴بازیابی متن گزینشی (کاندیدا)
۳۴تشخیص مرز متنی
۳۵فراپردازش
۳۶روش تجزیه مقدار منفرد
۳۸نقشه های خود-آرایی (Kohonen) [۱]
۳۸ماتریس فاصله یکپارچه
۳۹برچسب زنی نقش معنایی [۴]
۴۱درهم سازی جملات [۵]
۴۴مدل و نمونه [۶]
۴۵متن تکراری [۷]
۴۸فصل سوم : سرقت ادبی میان-زبانی
۴۹انواع روش های شناسایی سرقت ادبی میان زبانی
۴۹سیستم های مبتنی بر لغت
۵۰سیستم های مبتنی بر فرهنگ لغت
۵۰سیستم های مبتنی بر مجموعه های داده قابل مقایسه
۵۱سیستم های مبتنی بر مجموعه های داده موازی
۵۱سیستم های مبتنی بر ترجمه ماشینی [۹]
۵۱توصیف برخی از روش های مورد استفاده برای تشخیص سرقت ادبی میان زبانی
۵۱[۱۰] Ferret
۵۲طرح سیستم Ferret
۵۴N-گرم ها [۱۱]
۵۸فصل چهارم : مرور کوتاهی بر سرقت ادبی در صفحات وب
۵۹مقدمه [۱۲]
۶۰تحقیقات انجام شده
۶۴فصل پنجم : سرقت ادبی در کدهای کامپیوتری

۶۴	سرقت ادبی در علوم کامپیوتر
۶۵	مطالب سرقت شده در کدها [۱۳]
۶۸	انواع سیستم‌های شناسایی سرقت ادبی
۶۸	ابزارهای شناسایی سرقت ادبی کد منبع
۶۹	سیستم‌های مبتنی بر فینگرپرینت (اثر انگشت)
۷۲	روش‌های مقایسه محتویات
۷۲	الگوریتم‌های انطباق رشته
۷۸	الگوریتم‌های انطباقی پارامتری سازی شده
۷۹	الگوریتم‌های مقایسه درخت‌های تجزیه [۱۶]
۸۳	فهرست روش‌های شناسایی کدهای کپی شده [۱۳]
۸۷	فصل ششم : نتیجه گیری و تحقیقات آینده
۸۸	نتیجه گیری
۸۸	تحقیقات آینده
۹۰	منابع

فهرست جداول

۳۱	جدول ۱ : مزایا و معایب روش‌های سرقت ادبی درون زبانی
۴۳	جدول ۲: مثال
۵۵	جدول ۳: فهرست تری-گرم‌ها برای متن ۱
۵۵	جدول ۴: فهرست تری-گرم‌ها برای متن ۲
	جدول ۵: نتایج مفید توصیف شده. ستون Passage شامل متن‌ها است، N تعداد کلی تری-گرم‌ها در
۵۶	متن است، M تعداد تری-گرم‌های منطبق می‌باشد، و R ضریب تشابه است.
۵۷	جدول ۶ : مزایا و معایب روش‌های سرقت ادبی میان - زبانی

- جدول ۷: روش‌های رایج مورد استفاده برای شناسایی سرقت ادبی در کدهای کامپیوتری ۸۵
- جدول ۸: مزایا و معایب روش‌های تشخیص سرقت ادبی در کد نویسی ۸۶

فهرست تصاویر

- تصویر ۱: نمونه‌ای از یک مورد سرقت ادبی و نمایش n -گرم کلمات کلیدی (۸-گرم مشترک متون با خط‌ها نشان داده شده‌اند) ۳۳
- تصویر ۲: الگوریتم مربوط به تشخیص مرزهای متنی ۳۵
- تصویر ۳: تجزیه مقدار منفرد کاهش یافته به k ۳۷
- تصویر ۴: ساختار روش SRL ۴۱
- تصویر ۵: طبقه بندی انواع سرقت ادبی متنی، به همراه روش‌های شناسایی آن‌ها [۸] ۴۹
- تصویر ۶: سناریو ها و پاسخ‌ها ۶۷

خلاصه

سرقت ادبی از نقطه نظر «دزدی مالکیت معنوی» قدمتی به میزان تاریخ فعالیت‌های هنری و تحقیقاتی

انسان‌ها دارد. با این حال، دسترسی آسان به وب، پایگاه‌های داده ای بزرگ و به‌طور کلی ارتباطات راه دور،

باعث تبدیل سرقت ادبی به یک مشکل بزرگ برای ناشران، محققین و موسسات آموزشی شده است.

سرقت ادبی یک مشکل روزافزون در مؤسسات آکادمیک می‌باشد؛ و معمولاً به صورت کپی برداری از اثر

فردی دیگر (مثلاً از دانشجویان دیگر و یا منابع دیگری مانند کتاب‌های درسی)، و عدم اشاره به منبع مطالب

تعریف می‌شود (کاسما و جوی، ۲۰۰۸). وفور منابع آنلاین موجود، فاکتوری است که تأثیرات زیادی بر

افزایش وقوع سرقت ادبی در محیط‌های آکادمیک می‌گذارد زیرا استفاده از آثار دیگران برای دانشجویان

آسان‌تر شده است.

در این مقاله ما ابتدا در فصل ۱ به معرفی برخی از تحقیقات انجام شده، سوابق بررسی سرقت ادبی و

تاریخچه آن می‌پردازیم. پس از آن تحقیقات انجام شده از نظر دسته بندی روش‌های شناسایی

راتوضیح می‌دهیم. سپس می‌گوییم که سرقت ادبی چیست و چه اهمیتی دارد. دسته بندی های سرقت ادبی

در انتهای این فصل می‌آیند. فصل دوم به سرقت ادبی درون زبانی می‌پردازد و در فصل سوم سرقت ادبی

میان زبانی و روشهای آن را بررسی میکنیم. در فصول چهارم و پنجم بترتیب سرقت ادبی در صفحات وب و

سرقت ادبی در میان کدهای کامپیوتری، به همراه اهمیت و روشهای تشخیص آن می‌آیند. در نهایت یک

نتیجه گیری کلی از بحث در فصل آخر ارائه میگردد.

مقدمه

سرقت ادبی در مؤسسات آکادمیک معمولاً به صورت کپی برداری از اثر فردی دیگر (یعنی دانشجویان یا منابع دیگری مثل کتاب‌ها و مقالات) و عدم اشاره به نام منبع اثر (یعنی ارائه دهنده مطالب اصلی) تعریف می‌شود. سرقت ادبی بدون توجه به اینکه عمدی باشد یا غیرعمدی، جرم محسوب می‌شود.

هاناباس سرقت ادبی را به صورت «استفاده غیر مجاز و یا تقلید نزدیک از ایده‌ها و زبان/بیان فردی دیگر» تعریف می‌کند. در حوزه آثار آکادمیک، سرقت ادبی می‌تواند دامنه‌ای از بیان چند جمله از یک اثر بدون اشاره به نویسنده یا گوینده آن تا کپی برداری از کل سند را شامل شود. سرقت ادبی یک جرم آکادمیک است و یک جرم حقوقی محسوب نمی‌شود، و به صورت قوانین و مقررات سازمانی مطرح می‌گردد. بدین ترتیب، مفهوم سرقت ادبی در میان سازمان‌ها و مؤسسات بر مبنای قوانین و مقررات آن‌ها متفاوت است. همه دانشگاه‌ها، سرقت ادبی را شکلی از کلاهبرداری یا سوءرفتار آکادمیک می‌دانند، اما قوانین و مقررات آن‌ها برای کنترل موارد مذنون به سرقت ادبی متفاوت است و جرایم اعمال شده بر این کلاهبرداری مبتنی بر فاکتورهایی مانند شدت جرم و اقرار فرد متخلف به جرم خود می‌باشد. در میان مؤسسات آکادمیک، جرایم متفاوتی اعمال می‌شود، که می‌تواند از نمره صفر در درس مرتبط با اثر سرقت شده، معدوم نمودن اثر، و یا در موارد جدی‌تر، اخراج از مؤسسه را در بر بگیرد.

جالب است که فلینت، تحقیقی در میان ۲۶ فرد آکادمیک انجام داد تا مفهوم سرقت ادبی در میان دانشجویان را از آن‌ها جویا شود. افرادی که در تحقیقات و حضور پیدا کردند در دپارتمان‌ها و دانشکده‌های مختلف یک دانشگاه کار می‌کردند. او دریافت که کارمندان دارای «تعریف شخصی» خود از سرقت ادبی هستند. تعاریف موردنظر کارمندان تحت تأثیر تجارب پیشین در موارد مرتبط با سرقت ادبی در آموزش عالی بود و همچنین مشخص شد که این تعاریف با سیاست‌های سازمان مورد نظر در خصوص سرقت ادبی همخوانی ندارد. این ناهماهنگی‌ها در تعریف سرقت ادبی در میان افراد آکادمیکی که در یک مؤسسه کار می‌کنند، نشان‌دهنده ناهماهنگی در دنبال کردن سیاست دانشگاه در هنگام شناسایی موارد سرقت ادبی است. این مسئله می‌تواند به کار برد ناهماهنگ سیاست‌ها در میان افراد آکادمیک و همچنین اعمال جرایم/واکنش‌های متفاوت در برابر این موارد گردد. همچنین این مسئله می‌تواند باعث سردرگمی در میان دانشجویان پیرامون موضوع شود که سرقت ادبی در چه موردی اعمال می‌شود و در چه موردی که اعمال نمی‌شود.

سیستم‌های کامپیوتری و اتوماتیک تشخیص سرقت ادبی برای شناسایی این موارد در آثار مختلف به وجود آمده‌اند، و تأثیرگذاری این سیستم‌ها متکی بر نوع سرقت ادبی است که برای شناسایی آن به کار گرفته می‌شوند. این سیستم‌ها مزایای بسیاری از نظر صرفه‌جویی در زمان و فعالیت‌های افراد آکادمیک برای فرایند تشخیص دارم تشخیص کامپیوتری سرقت ادبی به این دلیل بر دو دهه گذشته مورد توجه افراد آکادمیک و محققین قرار گرفته است که استفاده از این ابزارها می‌تواند بار کاری آکادمیک را به واسطه اتوماتیک ساختن فرایندهای مقایسه و ارائه مطالب مشابه به طور قابل توجهی کاهش دهد.

فصل اول : کلیات

تحقیقات مرتبط و تاریخچه شناسایی سرقت ادبی

تحقیقات اولیه در مورد ارزیابی سیستم‌های شناسایی سرقت ادبی اساساً بر روی توصیف نقاط قوت و ضعف

سیستم‌های شناسایی متمرکز بودند (مثلاً کلاف، ۲۰۰۰؛ لنکستر و کالوین، ۲۰۰۴).

استفاده از ابزارهای کامپیوتری شناسایی سرقت ادبی همچنین با یک‌سری مسائل حقوقی و اخلاقی نیز

مرتبط هستند (مثلاً فاستر، ۲۰۰۲). این مسائل هم به واسطه نقایص تکنولوژیکی الگوریتم‌های شناسایی

سرقت ادبی (به‌عنوان مثال یک سیستم ممکن است تحقیقات یک دانشجو را به اشتباه سرقت ادبی

تشخیص دهد) و هم به واسطه سوءتفاهم در خصوص نقش نرم‌افزارهای شناسایی سرقت ادبی در فرایندهای

آموزشی پدید می‌آیند.

کاکانون و موزگووی (۲۰۱۰) یک ارزیابی سیستماتیک از هشت سیستم موجود آکادمیک و تجاری مرتبط با

شناسایی سرقت ادبی را برای متون دانشجویی به انجام رساندند. سیستم‌های مورد ارزیابی در تحقیقات آن‌ها

عبارت بودند از AntiPlagiarist (۲۰۱۰, ACNP Software)، EVE^۲ (Canexus، ۲۰۱۰)، Plagiarism-Finder

(Mediaphor، ۲۰۱۰)، SafeAssignment (Sciworth، ۲۰۱۰)، SeeSources (۲۰۱۰)، Joy & Lock) Shrlock

(۱۹۹۹)، Turnitin (iParadigms، ۲۰۱۰) و WCopyFind (BloomField، ۲۰۱۰). نتیجه اصلی تحقیقات آن‌ها آن

بود که سیستم‌های کنونی شناساگر دارای نقایص زیادی هستند که می‌توان آن‌ها را به دو دسته عمده زیر

تفکیک کرد:

- نقایص موجود در اجرای یک سیستم شناساگر بخصوص (مثلاً مسائل مرتبط با سهولت کاربری

سیستم)؛

- مسائل ناشی از محدودیت‌های تکنولوژی‌های موجود برای شناسایی سرقت ادبی.

بنت (۲۰۰۵) به مطالعه فاکتورهای تحریک کننده دانشجویان برای سرقت ادبی پرداخت و دریافت که

«ابزارها و فرصت‌ها» یکی از فاکتورهای موجود است. براساس این تحقیق این حقیقت که منابع به راحتی در

اینترنت در دسترس هستند، امکان دستیابی سریع و ساده به حجم زیادی از اطلاعات از منابع مختلف را

فراهم می‌آورد. همچنین، بسیاری از سایت‌های اینترنتی نیز وجود دارند که مقالات آماده را در اختیار

دانشجویان قرار می‌دهند. سهولت دستیابی به مطالب از منابع آن‌لاین و استفاده از آن‌ها در تحقیقات

آکادمیک، باعث افزایش شدید تعداد تحقیقات مرتبط با سرقت ادبی شده است (نیکسون و کسپرزاک ۲۰۰۴؛

نادلسون، ۲۰۰۷؛ اسکالون و نیومن، ۲۰۰۲).

سرقت ادبی چیست

«سرقت ادبی عبارتست از در اختیار گرفتن آثار فردی دیگر و انتقال آن‌ها به عنوان اثر شخصی. این جرم

ارتباط نزدیکی را با جعل و دزدی ادبی دارد- عملکرد حالی که به معنای نقض قوانین کپی‌رایت می‌باشند»:

دائرةالمعارف بریتانیکا.

سرقت ادبی در حقیقت یکی از جرایم الکترونیکی، مانند هک کامپیوتری، ویروس‌های کامپیوتری، اسپم،

نقض قوانین کپی‌رایت و غیره، می‌باشد. سرقت ادبی را می‌توان در اختیار گرفتن یا تلاش برای در اختیار

گرفتن یا استفاده (به صورت کلی یا جزئی) آثار فردی دیگر، بدون اشاره و ارجاع به او به‌عنوان مالک اثر

مطرح کرد. این کار می‌تواند به صورت یک copy و paste مستقیم، به واسطه اصلاح و تغییر برخی از لغات

متن اصلی که کتاب‌های اینترنتی، مجلات، روزنامه‌ها، تحقیقات، ژورنال‌ها، اطلاعات یا ایده‌های شخصی

حاصل می‌شوند، باشد.

تعاریف بسیاری از موارد تشکیل‌دهنده سرقت ادبی وجود دارد که ما به برخی از آن‌ها نگاهی می‌اندازیم. با

این حال بر اساس منابع تحقیقاتی ارائه‌شده در Plagiarism.org، مواردی که فوراً به ذهن می‌آیند را می‌توان

به‌صورت زیر ارائه کرد:

- مطرح کردن اثر فردی دیگر به عنوان اثر شخصی

- کپی کردن کلمات یا ایده‌ها از نظر فردی دیگر بدون اشاره به منبع
- ارائه اطلاعات ناقص در مورد منبع یک نقل قول
- تغییر کلمات و در عین حال کپی کردن ساختار جمله یک منبع بدون اشاره به آن
- کپی کردن تعداد زیادی از کلمات یا ایده‌ها از منبع به طوری که اکثریت اثر شما را تشکیل دهد، خواه با ارجاع (اشاره به منبع) همراه باشد یا خیر [Plagiarism.org, ۲۰۰۶]

اهمیت سرقت ادبی

در برخی از سازمان‌های آکادمیک مانند دانشگاه‌ها، دانشکده‌ها و مؤسسات، تشخیص و جلوگیری از سرقت ادبی به یکی از چالش‌های آموزشی تبدیل شده است، زیرا اغلب دانشجویان و محققین در هنگام انجام پروژه‌ها و فعالیت‌های محول شده به آن‌ها، به این روش تقلب روی می‌آورند. این مسئله ناشی از دسترسی پذیری منابع اینترنتی است. با دسترسی به این منابع، آن‌ها به راحتی می‌توانند با استفاده از یکی از موتورهای جستجو به دنبال موضوعات و تحقیقات گوناگونی بگردند بدون آن که نامی از مالک سند آورده شود. بنابراین برای همه حوزه‌های آکادمیک لازم است که از نرم‌افزارهای تشخیص سرقت ادبی در جهت جلوگیری یا حذف تقلب، کپی برداری و تغییرات سندهای دیگران توسط دانشجویان و افراد دیگر استفاده کنند.

برخی از انواع فعالیت‌های مرتبط با سرقت ادبی به راحتی می‌توانند با استفاده از نرم‌افزارهای تشخیص سرقت ادبی که اخیراً بوجود آمده‌اند و در بازار یا اینترنت موجود می‌باشند، کشف شوند. سرقت ادبی تنها

مورد استفاده دانشجویان نیست بلکه برخی از محققین و اعضای کارمندان مؤسسات آکادمیک نیز از سرقت ادبی در مقالات منتشره خود به طور مستقیم یا غیرمستقیم استفاده می‌کنند.

برنامه‌های کامپیوتری زیادی برای تشخیص سرقت ادبی به کار گرفته می‌شود و ابزارهای شناساگر گوناگونی توسط محققین به وجود آمده‌اند، اما آن‌ها همچنان دارای محدودیت‌هایی هستند و نمی‌توانند به درستی تشخیص دهنده و اثبات کننده سرقت ادبی باشند؛ بلکه صرفاً شباهت‌ها را نشان می‌دهند. با این حال بسیاری از دانشگاه‌ها و مراکز تحقیقاتی هنوز هیچ اقدامی را برای مقابله با سرقت ادبی انجام نداده‌اند و این مسئله باعث رشد بیشتر سرقت ادبی شده است. علیرغم این موارد، حتی با استفاده از نرم‌افزارهای جدید شناساگر نیز نمی‌توان سرقت ادبی راه ۱۰۰٪ محو کرد.

کپی رایت و جنبه‌های حقوقی استفاده از اسناد منتشر شده نیز می‌توانند تحت پوشش نرم‌افزارهای شناسایی سرقت ادبی قرار گیرند تا مشخص شود که آیا یک فرد از اسناد دیگران به صورت قانونی استفاده کرده است یا غیر قانونی و اینکه آیا فرد مجوز لازم را از طرف مالک برای استفاده از این سند داشته است یا خیر.

شناسایی سرقت ادبی همچنین یکی از مهم‌ترین مسائل برای سازمان‌ها، مراکز تحقیقاتی و کنفرانس‌ها است؛ آن‌ها از ابزارهای پیشرفته شناسایی سرقت ادبی برای اطمینان از این موضوع استفاده می‌کنند که اسناد آن‌ها مورد سرقت ادبی قرار نگیرند، و در نتیجه حق کپی رایت ناشران محفوظ بماند.

یک تحقیق (منتشر شده در ژوئن ۲۰۰۵) که به صورت بخشی از پروژه ارزیابی مرکز یکپارچگی آکادمیک انجام شد، نشان داد که ۴۰٪ از دانشجویان به انجام فعالیت‌های مرتبط با سرقت ادبی اقرار کرده‌اند؛ در

حالی که این مسئله در سال ۱۹۹۹، ۱۰٪ بود [CAI، ۲۰۰۵]. یک مطالعه جمعی دیگر که توسط استاد

دانشگاه Rutgers در سال ۲۰۰۳ انجام شد، گزارش داد که ۳۸٪ از دانشجویان در سرقت ادبی آن لاین

مشارکت داشتند [روتگار، ۲۰۰۳]. این ارقام هشداردهنده بیانگر افزایش تدریجی سرقت ادبی می باشد. نسل

جدید نسبت به گذشته از تکنولوژی آگاه تر هستند. سرقت ادبی هم اکنون محدود به یک copy و paste کردن

صرف نیست؛ تکنولوژی های مترادف سازی و ترجمه در حال ایجاد ابعاد جدیدی در سرقت ادبی هستند.

هم اکنون سرقت ادبی مهم ترین سوء رفتار آکادمیک محسوب می شود؛ آکادمی ها و مؤسسات سراسر جهان

در حال شروع فعالیت هایی برای آموزش به دانشجویان و مدرسین در جهت توصیف و تحلیل انواع سرقت

ادبی و چگونگی اجتناب از آن می باشند.

نادلسون (۲۰۰۷) تحقیقی را در میان ۷۲ فرد آکادمیک در خصوص مسائل مرتبط با سوء رفتار آکادمیک

انجام داد و گزارش کرد که ۵۷۰ نمونه از سرقت ادبی مظنون مشاهده شده است. اغلب موارد گزارش شده

شامل «سرقت ادبی تصادفی/غیر عمدی» بوده اند که در این میان ۱۳۴ مورد مربوط به دانشجویان دوره

لیسانس و ۳۹ مورد مربوط به دانشجویان فارغ التحصیل بود. بعلاوه، افراد آکادمیک گزارش کردند که تعداد

زیادی از دانشجویانی را سراغ دارند که مقالات کپی شده از اینترنت را ارائه کرده اند. همچنین مواردی از

«سرقت ادبی هدفمند»، «تقلب های آزمون کلاسی» و «تقلب های آزمون های خانگی» نیز گزارش شدند.

بعلاوه سرقت ادبی مشکل بزرگی در دوره های برنامه نویسی است. کالوین (۲۰۰۱) تحقیقی را از سرقت ادبی

کد منبع انجام داد و در آن داده هایی را از ۵۵ دانشکده کامپیوتر در بریتانیا به دست آورد. او دریافت که

۵۰٪ از ۲۹۳ فرد آکادمیک مورد بررسی عقیده داشتند که سرقت ادبی در سال‌های اخیر افزایش یافته است.

بعلاوه، ۲۲ نفر از ۴۹ پاسخ دهنده برآوردهایی از ۲۰٪ تا ۵۰٪ را در خصوص دانشجویان مشارکت کننده در

فعالیت‌های سرقت ادبی در دوره‌های ابتدایی برنامه‌نویسی در ذهن داشتند.

مقیاس مشکل

یک بررسی اولیه در میان ۳۸۰ دانشجوی دوره کارشناسی که توسط هاینز، دیکهوف، لاف و کلارک در سال

۱۹۸۶ انجام شد نشان داد که هر چند حدود نیمی از دانشجویان به انواع مختلف تقلب می‌کنند، اما تنها

۱٫۳٪ از این موارد، تشخیص داده می‌شود. بررسی‌های اخیر نشان داده‌اند که در دو دهه گذشته، سرقت ادبی

در میان دانشجویان به دلیل سهولت دسترسی و اشتراک گذاری پاسخ‌های تکالیف و نیز مطالب تحقیقاتی،

بسیار رایج شده است.

سرقت ادبی در تکالیف برنامه‌نویسی یک مشکل روزافزون در محیط‌های آکادمیک است. کد منبع را می‌توان

به طرق مختلف مانند اینترنت، بانک‌های کد منبع و کتاب‌های راهنما به دست آورد. در خصوص موارد

مظنون به سوءرفتار آکادمیک، تحقیقات مشخص کردند که به دلیل مشکلات روش‌های رسمی دانشگاه‌ها،

افراد آکادمیک ترجیح می‌دهند که مسائل دخیل در سرقت ادبی را به صورت غیررسمی رفع و رجوع کنند.

این مشکلات اساساً ناشی از دو مسئله است-افراد آکادمیک احساس می‌کنند که شواهد کافی را برای گزارش

این سرقت‌ها ندارند، و اینکه پیگیری رسمی موارد سوءرفتار آکادمیک می‌تواند وجهه آن‌ها را به عنوان افراد

آکادمیک به صورت منفی تحت تأثیر قرار دهد. بعلاوه، کیت-اشپینگل دریافته‌اند که افراد آکادمیک عموماً

سیاست‌های دانشگاه را در مورد سرقت ادبی دنبال نمی‌کنند زیرا از مواجهه با دانشجویان و عدم پشتیبانی

شیوه‌های دانشگاهی بیم دارند. آن‌ها همچنین دریافته‌اند که افراد آکادمیک با توجه به تأثیراتی که ممکن است روش‌های رسمی پیگیری سرقت ادبی بر دانشجویان داشته باشد، با آن‌ها با ملایمت بیشتری رفتار می‌کنند.

دسته‌بندی‌های سرقت ادبی

دسته‌های گسترده‌تر سرقت ادبی شامل موارد زیر می‌باشند:

- تصادفی: به دلیل عدم اطلاع از چگونگی سرقت ادبی و عدم درک چگونگی استفاده از روش‌های ارجاع (اشاره به منبع) مورد استفاده در یک سازمان
- غیرعمدی: گستردگی اطلاعات موجود می‌تواند بر ایجاد شباهت‌هایی میان تفکر و ایده‌های به دست آمده از مطالب مکتوب یا شفاهی افراد اثرگذار باشد
- عمدی: عملکرد تعمدی کپی برداری کلی یا جزئی از آثار فردی دیگر بدون اشاره به منبع اثر
- سرقت ادبی شخصی: استفاده از آثار منتشرشده شخصی به شکلی دیگر بدون ارجاع به اثر اصلی

[Wikipedia:Plagiarism ۲۰۰۶] [بیزلی، ۲۰۰۶].

به طور دقیق‌تر، و از جنبه روش کار، سرقت ادبی می‌تواند اشکال مختلفی به خود بگیرد؛ که شامل موارد زیر می‌باشد.

- کپی برداری کلمه به کلمه، که شامل کپی برداری مستقیم جملات یا مجموعه‌ای از جملات از آثار دیگران بدون ارجاع به منبع اصلی می‌باشد.

- تغییر الفاظ، که شامل بازنویسی نزدیک (تنها تغییر برخی از کلمات بدون تغییرات کلی) متن مکتوب بدون ارجاع مناسب به منبع اثر می‌باشد.
- سرقت ادبی از منابع ثانویه، که شامل ارجاع و اشاره به منبع ثانویه یک متن بدون جستجو به دنبال منبع اصلی می‌باشد.
- سرقت ادبی شکل یک منبع زمانی اتفاق می‌افتد که ساختار یک آرگومان در یک منبع بدون استفاده سیستماتیک از ارجاعها از یک منبع ثانویه انجام می‌شود. این گونه سوء رفتار شامل جستجوی مرجع ها و دنبال کردن ساختاری مشابه منبع ثانویه می‌باشد.
- سرقت ادبی ایده‌ها، که شامل استفاده از ایده‌های مطرح‌شده در یک متن منبع بدون هیچ گونه اشاره مستقیم به کلمات یا شکل منبع می‌باشد.
- سرقت ادبی مشهود یا مالکیتی، یعنی در اختیار گرفتن اثر دیگران و معرفی آن بعنوان اثر خود.

تحقیقات انجام شده از نظر دسته بندی روش‌های شناسایی

ردیاب‌ها^۱

بر مبنای تحقیقات الظهرانی پیرامون روش‌های تشخیص سرقت ادبی، روش‌های رایج مبتنی هستند بر کاراکترها برای مقایسه اسناد مذنون با اسناد اصلی. رشته همسان می‌تواند به طور دقیق یا جزئی با استفاده از روش انطباقی کاراکتر تشخیص داده شود. لیون مقایسه متنی را بر مبنای n -گرم های کلمات انجام می‌دهد؛ و با ارجاع به آن‌ها، متن مورد نظر به دو مجموعه از تری-گرم ها تقسیم می‌شود تا مورد مقایسه

^۱ Fingerprints

قرار گیرد. میزان تری-گرم های مشترک به منظور بررسی موارد بالقوه سرقت ادبی، شناسایی می‌شود. کنگ جمله را به عنوان واحد مقایسه در نظر می‌گیرد تا شباهت‌های محلی میان آن‌ها را بررسی کند. او تمایزی را میان کپی برداری دقیق جملات، درون گذاری کلمه، حذف کلمات و تعویض آن‌ها با کلمات هم معنی قایل می‌شود. هاینتره، برودرو کروسزتی یک روش ردیابی را برای یافتن انطباق رشته‌ای و تشخیص سرقت ادبی بر مبنای نسبت ردیاب های رایج پیشنهاد کردند. این روش‌ها نتایج خوبی را به دست می‌دهند، اما نمی‌توانند بخش‌های حاوی کلمات تعویض شده با کلمات هم معنی یا متفاوت را تشخیص دهند.

شباهت رشته‌ای^۲

برین یک سیستم تشخیص سرقت ادبی را با نام COPS (سیستم محافظت در برابر کپی) از پروژه کتابخانه دیجیتال استنفرد ایجاد کرد، که می‌تواند تداخل اسناد را بر مبنای جملات و انطباق رشته‌ای تشخیص دهد. نقص اصلی این روش، ناتوانی در بررسی کلمات منفرد و در نظر گرفتن کل جمله به عنوان یک بخش منفرد است. نقایص روش COPS توسط شیواکومار و گارسیا-مولینا برطرف شد و روش جدید به دست آمد با نام روش آنالیز کپی استانفورد (SCAM) تا بتواند COPS را با استفاده از مدل تناوب نسبی (RFM) برای نشان دادن زیرمجموعه‌های کپی برداری شده، بهبود بخشد. RFM عبارتست از یک معیار شباهت نامتقارن برای تشخیص سرقت ادبی. مزیت اصلی SCAM آن است که می‌تواند شباهت‌های متداخل را میان اجزای جمله‌ها تشخیص دهد، اما عبارات بسیاری وجود دارند که می‌توانند در مقایسه تقسیمات میان اسناد، همراه‌کننده باشند. روش پیشنهادی توسط هانت و سزیمانسکی و اقتباس شده توسط چو و سلیم، بلندترین زیر-دنباله

^۲ String Similarity

مشترک (LCS) نام دارد. روش LCS یکی از شیوه‌های مورد استفاده در ROUGE است که در حقیقت یک روش ارزیابی خلاصه معروف می‌باشد. با داشتن دو دنباله X و Y ، بلندترین زیر-دنباله مشترک برای آن‌ها می‌تواند زیر دنباله مشترک دارای بلندترین ماکزیمم باشد. وایت و جوی یک الگوریتم جدید را برای تشخیص سرقت ادبی پیشنهاد کردند که می‌تواند تداخل‌های پیچیده (مانند تغییرات کلمات، بازآرایی، یکپارچه سازی، و تفکیک جملات) و نیز کپی برداری مستقیم را تشخیص دهد.

روش‌های مبتنی بر ساختار^۳

توجه به این نکته اهمیت دارد که همه مطالعات پیش‌گفته، روش‌های مبتنی بر کاراکتر را توصیف کردند. در حقیقت همه این روش‌ها روی ویژگی‌های کلمات متن در یک سند متمرکز بودند. با این حال بسیاری از مطالعات، روش‌های مختلفی را در حوزه تشخیص سرقت ادبی بر مبنای ویژگی‌های ساختاری متن موجود در سند، مانند سربرگ، تعداد بخش‌ها، پاراگراف‌ها و منابع پیشنهاد نموده‌اند. ویژگی‌های ساختار درختی یکی از مطالعات اخیر با تمرکز بر ویژگی‌های ساختاری است. این یک مدل از نقشه‌های خود-آرای چند لایه (ML-SOM) برای اسناد متنی می‌باشد. چو و رحمان از یک نمایش ساختار درختی با ML-SOM برای تشخیص سرقت ادبی و ارزیابی اطلاعات استفاده کردند. ایده آن‌ها مبتنی بر دو مرحله بود، لایه بالا و لایه پایین. لایه بالا نشان‌دهنده دسته‌بندی و ارزیابی اسناد است در حالی که لایه پایین از یک ضریب شباهت کسینوسی برای تعیین شباهت‌های میان متون در نتیجه تشخیص سرقت ادبی استفاده می‌کند.

^۳ Structure-Based Methods

روش‌های دسته‌ای^۴

روش دسته‌بندی سند یکی از روش‌های بازیابی اطلاعات است که در بسیاری از حوزه‌ها مانند خلاصه سازی متن، دسته‌بندی متن، و تشخیص سرقت ادبی مورد استفاده قرار می‌گیرد. از آن برای بهسازی داده‌های بازیابی شده با استفاده از کاهش زمان جستجو در موقعیت سند برای خلاصه سازی متن و کاهش زمان مقایسه در تشخیص سرقت ادبی استفاده می‌شود. شیوه دیگری که توسط سی و همکاران، و زینی پیشنهاد شده، از کلمات خاص (کلمات کلیدی) برای یافتن دسته‌های مشابه میان اسناد استفاده می‌کند. زینی نیز اشاره کرد که روش‌های ردیابی اساساً مبتنی بر استفاده از k -گرم‌ها است. از آنجا که فرایند ردیابی تقسیم کننده اسناد به گرم‌های (grams) دارای طول k هستند، اما ردیابی‌های سند را می‌توان به منظور تشخیص سرقت ادبی، مقایسه کرد. روش‌های مبتنی بر ردیابی برای دسته‌بندی اسناد برای خلاصه سازی مجموعه‌ای از اسناد و ایجاد یک مدل ردیابی برای آن‌ها مورد استفاده قرار گرفتند.

مدلهای نحوی^۵

الهادی و التوبی یک روش تشخیص کپی برداری را برای ساختارهای نحوی اسناد پیشنهاد کردند. این روش نگاهی دارد به استفاده از برچسب‌های اقسام نحوی جمله (PoS) برای نمایش ساختار متن به صورت مبنایی برای مقایسه و آنالیز بیشتر. این روش می‌تواند اسناد را با استفاده از برچسب‌های PoS مرتب و رتبه‌بندی کند. الهادی و التوبی روش خود را برای محاسبه شباهت میان اسناد و رتبه بندی آن‌ها بر مبنای مرتبط ترین اسناد استخراج شده ارتقا دادند.

^۴ Cluster Based Methods
^۵ Syntactic Models

برخی از مطالعات مانند تحقیق کروتشف و سربیان ترکیبات نحوی دو متن را بر مبنای الگوریتم فاصله نرمال سازی شده Lempel-Ziv (LZ) متراکم سازی کردند و شباهت‌های میان اطلاعات توپولوژیکی مشترک ارائه شده توسط متراکم ساز را محاسبه نمود. این روش می‌توانست شباهت‌های میان اسناد متنی را حتی در صورتی تشخیص دهد که آن‌ها دارای الفاظ متفاوت می‌باشند. بعلاوه، این روش و سایر روش‌هایی مانند چیزی که توسط الهادی و همکارانش پیشنهاد شد، مبتنی بودند بر کاهش متن با استفاده از نشانه سازی و حذف کلمات تکراری و تنها مجموعه‌های کوچک‌تری از برچسب‌های نحوی را مدنظر قرار می‌دهند.

روش‌های چند زبانی^۶

یک روش چند زبانی برای تشخیص اسناد مذنون توسط گروزیا و پوپسکو پیشنهاد شد. شباهت میان اسناد اصلی و مذنون با استفاده از یک مدل آماری ارزیابی می‌شود. این مدل می‌تواند این احتمال را تعیین کند که یک سند مذنون با سند اصلی مرتبط است یا خیر، بدون توجه به ترتیب عبارات در دو سند. روش آن‌ها با یک مجموعه داده متنی در انگلیسی و اسپانیایی ترکیب شده با شباهت‌های میان زبانی را تشخیص دهد. این روش نیازمند ایجاد یک مجموعه داده میان زبانی است که همان‌طور که گروزیا و پوپسکو گفته‌اند، کار بسیار دشواری است.

روش‌های معنایی (صرفی)^۷

بسیاری از محققین کارهای زیادی را برای محاسبه شباهت‌های معنای میان کلمات و ارتباط میان اسناد با استفاده از WordNet انجام داده‌اند. معیارهای مبتنی بر اطلاعات می‌توانند شباهت معنایی میان دو کلمه را با

^۶ Cross-Language Methods
^۷ Semantic Methods

محاسبه درجه ارتباط میان آن‌ها با استفاده از اطلاعات یک دیکشنری یا فرهنگ لغت تشخیص دهند. در اینجا از روابط و سلسله‌مراتب یک فرهنگ لغت استفاده می‌شود که عموماً یک پایگاه داده لغوی ایجاد شده به صورت دستی مانند WordNet می‌باشد. تا آنجا که ما می‌دانیم، اسناد متنی می‌توانند به صورت یک گراف نمایش داده شوند. سرقت ادبی می‌تواند به واسطه انطباق گراف میان مفاهیم کلمات نشان داده شود. یک روش معنای تشخیص سرقت ادبی که توسط الظهرانی و سلیم پیشنهاد شد، از شباهت رشته‌ای مبتنی بر معنای فازی استفاده می‌کند. این روش به واسطه چهار مرحله اصلی انجام گرفت. مرحله اول عبارتست از پیش پردازش که شامل نشانه سازی، حذف کلمات مشابه و ریشه‌یابی. مرحله دوم عبارت است از بازیابی فهرستی از اسناد منتخب برای هر سند مذنون با استفاده از ضریب Jaccard و الگوریتم Shingling. سپس اسناد مذنون به صورت جمله به جمله با اسناد منتخب مقایسه می‌شوند. این مرحله دربرگیرنده محاسبه درجه شباهت فازی است که دامنه‌ای از ۰ برای دو جمله کاملاً متفاوت را ۱ برای دو جمله کاملاً یکسان دارد. دو جمله در صورتی علامت‌گذاری می‌شوند که دارای یک نمره شباهت فازی در بالای یک آستانه معین باشند. مرحله آخر، پردازش واپسین است که در آن جملات متعاقب به یکدیگر پیوند داده می‌شوند تا یک پاراگراف یا بخش منفرد را شکل دهند.

روش متفاوتی از تشخیص سرقت ادبی به صورت معنایی توسط چو و سلیم پیشنهاد شد. روش پیشنهادی می‌تواند شباهت میان اسناد مذنون و اصلی را بر مبنای مستندات جمله محاسبه کند. هر یک از این مستندات با استفاده از یک درخت تجزیه گر استانفورد (SPT) استخراج می‌شود. درجه شباهت میان

مستندات استخراج شده با استفاده از فرهنگ لغت WordNet محاسبه می‌شود. نقص این روش آن است که

نمی‌تواند همه بخش‌های جملات را تشخیص دهد بلکه تنها فاعل، فعل و مفعول را بررسی می‌کند.

فصل دوم : سرقت ادبی

درون زبانی

سرقت ادبی به نام زبانی در حقیقت چیزی است که آن را به صورت دریافت عبارات، مفاهیم و جملات از

متن دیگر و انتقال آن به متن خود در حیطه یک زبان منفرد تعریف می‌کنند. روش‌های بسیاری برای

تشخیص سرقت ادبی توسط محققین در سال‌های گذشته به وجود آمده‌اند شما در اینجا به برخی از آن‌ها

اشاره می‌کنیم که مرتبط با تشخیص ادبی در حیطه زبان منفرد هستند.

ابتدا، برخی از محققین از روش‌های تشخیص کپی متن «زبان طبیعی»^[۱۲] استفاده کردند که این روش در

دهه ۹۰ پدیدار شد و در آن سه روش تشخیص سرقت ادبی مطرح شد.

روش‌های مبتنی بر گرامر^[۱]

روش مبتنی بر گرامر یکی از مهم‌ترین روش‌های مورد استفاده برای تشخیص سرقت ادبی است که بر

ساختار گرامری اسناد تمرکز می‌کند و از یک شیوه انطباق رشته‌ای برای شناسایی و اندازه‌گیری شباهت

میان اسناد بهره می‌گیرد. روش‌های مبتنی بر گرامر برای تشخیص کپی‌های دقیق بدون هرگونه تغییرات،

مورد استفاده قرار می‌گیرند اما نمی‌توانند متون کپی شده و در عین حال اصلاح شده را تشخیص دهند. این

یکی از محدودیت‌های این روش‌ها است. در این زمینه می‌توان به روش‌های زیر اشاره کرد:

- هوانگ یک روش تشخیص برای صفحات وب را بر مبنای الگوریتم LCS (پیش گفته) از طریق یافتن

بزرگ‌ترین رشته مشترک میان دو صفحه برای محاسبه شباهت میان آن‌ها ارائه کرد.

[^] Grammer-based methods

- اشلاimer، ویلکرسون و آیکن از روش k گرم تداخلی برای دریافت hash های اسناد و به دست آوردن

ردیاب ها با کاهش تعداد hash ها در سند استفاده کردند و سپس از ردیاب برای هر سند به دست

آمده بهره گرفتند و نرخ های شباهت میان این اسناد را شمردند.

- Hashbreaking و DCT نیز از روش های گرامر-محور هستند؛ تنها تفاوت میان آن ها چگونگی دریافت

ردیاب های میان اسناد است.

روش معنایی^۹

روش معنایی نیز یکی از روش های مهم برای تشخیص سرقت ادبی است و مبتنی بر شباهت های میان اسناد

به واسطه مدل فضای برداری می باشد. این روش همچنین می تواند حشویات کلمات را در سند شمرده و

محاسبه کند و سپس از ردیاب های اسناد دیگر برای یافتن شباهت استفاده کند.

روش ترکیبی گرامر-معنایی

روش ترکیبی گرامر-معنایی یکی از مهم ترین روش ها در تشخیص سرقت ادبی برای زبان های طبیعی است.

این روش، که در دستیابی به نتایج بهتر و مناسب تر تأثیرگذاری خود را نشان داده است، برای متون کپی

شده، تغییر یافته، و بازنویسی شده ای مناسب است که نمی توان آن ها را از طریق روش های گرامر-محور و

نیز روش های معنایی بررسی کرد.

دوم آنکه برخی از روش ها از ساختارهای شاخص-محور بخصوصی استفاده کردند:

^۹ Semantic methods

- ملکولم و لین از سیستم تشخیص سرقت Ferret بهره گرفتند که مبتنی است بر تری-گرم های کلمات مشترک.

- باسیل متن ها را به صورت یک دنباله طول کلمات رمز گذاری کرد و از شاخص فاصله n -گرم مبتنی بر بردار جریان نزولی برای انتخاب بخش معیار (کاندیدا) استفاده کرد.

- کاسپرزاک شینگل های متنی مشترک را در فرایند پیش-گزینش مورد استفاده قرار داد و اشکربینین و بوتاکف از ردیاب های مبتنی بر hash برای بازیابی پخش گزینشی استفاده نمودند.

سوم، روش تشخیص سرقت ادبی خارجی :

روش خارجی تشخیص سرقت ادبی مبتنی است بر یک مجموعه داده مرجع متشکل از اسنادی که در آن ها متون ممکن است سرقت شده باشند. یک متن می تواند از پاراگراف ها ، یک بلوک از کلمات با اندازه ثابت، یک بلوک از جملات و غیره تشکیل شود. یک سند مذنون به واسطه جستجو برای متونی کپی برداری شده که در مجموعه داده مرجع قرار دارند، از نظر سرقت ادبی مورد بررسی قرار می گیرند. سپس یک سیستم خارجی سرقت ادبی این یافته ها را برای یک کنترلگر انسانی گزارش می کند و او تشخیص می دهد که آیا متون شناسایی شده مورد سرقت قرار گرفته اند یا خیر. یک راهکار برای این مسئله می تواند مقایسه هر متن در سند مذنون با هر متنی از یک سند در یک مجموعه داده مرجع باشد.

مزایا و معایب : مجموعه داده مرجع باید به حد کافی بزرگ باشد تا حداکثر میزان ممکن از متون سرقت شده یافت شوند. این موضوع باعث افزایش شدید زمان اجرا میشود.

نمونه‌هایی از تحقیقات انجام شده برای تشخیص سرقت ادبی خارجی عبارتند از:

- تشخیص اتوماتیک جهت‌گیری سرقت ادبی؛ از این روش برای تعیین جهت سرقت ادبی، استفاده شد. گروزیا و پوپسکو از تعمیمی از روش Encoplot برای این منظور بهره گرفتند و آن را روی مجموعه بزرگی از سرقت‌های ادبی مصنوعی آزمودند و نشان دادند که در بزرگ‌ترین مجموعه داده ای سرقت ادبی موجود تا به امروز، مشکل جهت‌گیری سرقت ادبی با دقت نسبتاً بالا (در حدود ۷۵٪) برطرف شد، اما آن‌ها این روش را در مورد زبان طبیعی نیازموندند.
- روش تشخیص اتوماتیک سرقت ادبی به شکل خارجی از شباهت‌های متنی استفاده کرد، این روش در مرحله پیش پردازش به کار گرفته شد. وانی و آدریانی اسناد بازیابی شده را به متون S تقسیم کردن و هر متن دارای ۲۰ جمله بود و سپس سرقت ادبی را با شناسایی تعداد لغات متداخل میان متون مذنون و مرجع شناسایی کردند.
- دوی، رائو، رام و آکیلاندسوار ی یک الگوریتم را برای شناسایی سرقت ادبی خارجی در رقابت PAN-۱۰ مطرح کردند. این الگوریتم دارای دو مرحله بود: مرحله اول شناسایی اسناد مشابه و بخش سرقت شده برای سند مذنون با مدل فضای برداری (VSM) و شاخص شباهت کسینوسی، و مرحله دوم شناسایی نواحی سرقت شده در متن مذنون با استفاده از نسبت Chunk. اما پیش پردازش اسناد انجام نشد.
- مارکوس موهر، کرن، زکمر و گرانیتر سیستم ترکیبی خود را برای رقابت PAN در CLEF ۲۰۱۰ مطرح نمودند. سیستم آن‌ها تشخیص سرقت ادبی را برای متون ترجمه شده و ترجمه نشده و نیز

متون اسناد سرقت شده داخلی انجام می‌داد؛ روش تشخیص سرقت ادبی خارجی به صورت یک

مسئله بازیابی اطلاعات فرمول بندی شد، و از فرا-پردازش شهودی برای دستیابی به نتایج شناسایی

نهایی استفاده کرد.

• زکمر و همکاران، یک مدل استاندارد از IR متنی را برای گزینش بخش کاندیدا به کار گرفتند. اسناد

مرجع در سطح جمله، شاخص گذاری شدند و جملات اسناد مذنون به عنوان عبارات جستجو به کار

گرفته شدند. شباهت از طریق شاخص کسینوس برآورد شد.

روش	مزایا/معایب [۲]
روش گرامر-محور	روشهای گرامر-محور برای تشخیص کپی های دقیقی مناسب هستند که در آنها هیچ تغییری در متن مشاهده نمی شود؛ اما نمی توانند متونی که شامل بازنویسی یا جابجایی کلمات هستند (اما معنای مشابهی دارند) را شناسایی نمایند.
روش معنایی	روش مبنایی مبتنی است بر سرقت ادبی غیر جزئی و از کل سند و نیز از فضای برداری برای انطباق میان اسناد استفاده می کند ؛ اما چنانچه سند به صورت جزئی مورد سرقت قرار گرفته باشد، نمی تواند نتایج مناسبی را به دست دهد و این یکی از محدودیت های روش مذکور است.
روش ترکیبی گرامر-معنایی	بصورتی که گفته شد روش ترکیبی برای تشخیص متون تغییر یافته از طریق بازنویسی یا جابجایی مناسب است که توسط روشهای گرامری

<p>صرف قابل تشخیص نیستند. همچنین این شیوه قادر است محدودیت های روش معنایی را نیز برطرف کند و موقعیت بخشهای سرقت شده را در سند تشخیص دهد.</p>	
--	--

جدول ۱: مزایا و معایب روش های تشخیص سرقت ادبی درون زبانی

نمونه‌هایی از روش‌های تشخیص سرقت ادبی درون زبانی^{۱۰}

تشخیص سرقت ادبی بر مبنای اطلاعات ساختاری [۳]

در این روش فرض می‌شود که D_x مجموعه‌ای از اسناد مذنون و D_s مجموعه‌ای از اسناد مرجع باشند.

اولین کار عبارت است از تعیین اینکه آیا یک سند مذنون مورد سرقت قرار گرفته است یا خیر. باید همه

منابع سرقت، مانند اسناد منبع (زیر مجموعه‌ای از D_s) و مرزهای دقیق متون سرقت شده در اسناد مذنون

و مرجع شناسایی شود. بعلاوه، بهتر است نمره‌ای به هر متن سرقت شده شناسایی شده اختصاص داده شود

که نشان دهنده درجه سرقت ادبی در آن است. این نمره را می‌توان برای مرتب کردن متن‌های شناسایی

شده از «کپی‌های دقیق» تا «متن‌های نسبتاً مرتبط» به کار گرفت.

نمایش متن

نمایش متون براساس روش پیشنهادی، مبتنی است بر n -گرم های کلمات کلیدی (SWNG). با داشتن یک

سند و فهرستی از کلمات کلیدی، متن به نمایشی از این کلمات کلیدی در سند تقلیل پیدا می‌کند. همه

نشانه‌های دیگر حذف می‌شوند.

در تصویر ۱، نمونه‌ای از نمایش SWNG را مشاهده می‌کنید. متن سمت راست، متن اصلی بوده و متن سمت

چپ، متنی است که مظنون به سرقت ادبی می‌باشد.

مزایا و معایب: این روش میتواند بازدهی بسیار بالایی را در مورد سرقت های ادبی دشوار بدست آورد که در

آنها تغییرات قابل توجهی در متن رخ داده است و کلمات و عبارات با مترادف های آنها جایگزین شده اند.

کارکرد این روش بسیار آسان است و نیازمند حداقل منابع و حداقل هزینه پردازش متن می باشد.

Suspicious passage:

This came into existence likely from the deviance in the time-period of the particular billet. As the premier is to be nominated for not more than a period of four years, it can infrequently happen that an ample wage, fixed at the embarkation of that period, will not endure to be such to its end.

Original passage:

This probably arose from the difference in the duration of the respective offices. As the President is to be elected for no more than four years, it can rarely happen that an adequate salary, fixed at the commencement of that period, will not continue to be such to its end.

SWNG representation:

[this, from, the, in, the, of, the, as] — [this, from, the, in, the, of, the, as]
 [from, the, in, the, of, the, as, the] — [from, the, in, the, of, the, as, the]
 [the, in, the, of, the, as, the, is] — [the, in, the, of, the, as, the, is]
 [in, the, of, the, as, the, is, to] — [in, the, of, the, as, the, is, to]
 [the, of, the, as, the, is, to, be] — [the, of, the, as, the, is, to, be]
 [of, the, as, the, is, to, be, for] — [of, the, as, the, is, to, be, for]
 [the, as, the, is, to, be, for, not] — [the, as, the, is, to, be, for, it]
 [as, the, is, to, be, for, not, a] — [as, the, is, to, be, for, it, can]
 [the, is, to, be, for, not, a, of] — [the, is, to, be, for, it, can, that]
 [is, to, be, for, not, a, of, it] — [is, to, be, for, it, can, that, an]
 [to, be, for, not, a, of, it, can] — [to, be, for, it, can, that, an, at]
 [be, for, not, a, of, it, can, that] — [be, for, it, can, that, an, at, the]
 [for, not, a, of, it, can, that, an] — [for, it, can, that, an, at, the, of]
 [not, a, of, it, can, that, an, at] — [it, can, that, an, at, the, of, that]
 [a, of, it, can, that, an, at, the] — [can, that, an, at, the, of, that, will]
 [of, it, can, that, an, at, the, of] — [that, an, at, the, of, that, will, not]
 [it, can, that, an, at, the, of, that] — [an, at, the, of, that, will, not, to]
 [can, that, an, at, the, of, that, will] — [at, the, of, that, will, not, to, be]
 [that, an, at, the, of, that, will, not] — [the, of, that, will, not, to, be, to]
 [an, at, the, of, that, will, not, to] —
 [at, the, of, that, will, not, to, be] —
 [the, of, that, will, not, to, be, to] —

SWNG representation:

[this, from, the, in, the, of, the, as]
 [from, the, in, the, of, the, as, the]
 [the, in, the, of, the, as, the, is]
 [in, the, of, the, as, the, is, to]
 [the, of, the, as, the, is, to, be]
 [of, the, as, the, is, to, be, for]
 [the, as, the, is, to, be, for, it]
 [as, the, is, to, be, for, it, can]
 [the, is, to, be, for, it, can, that]
 [is, to, be, for, it, can, that, an]
 [to, be, for, it, can, that, an, at]
 [be, for, it, can, that, an, at, the]
 [for, it, can, that, an, at, the, of]
 [it, can, that, an, at, the, of, that]
 [can, that, an, at, the, of, that, will]
 [that, an, at, the, of, that, will, not]
 [an, at, the, of, that, will, not, to]
 [at, the, of, that, will, not, to, be]
 [the, of, that, will, not, to, be, to]

تصویر ۱: نمونه‌ای از یک مورد سرقت ادبی و نمایش n-گرم کلمات کلیدی (۸- گرم مشترک متون با خط‌ها نشان داده شده‌اند)

بازیابی متن گزینشی (کاندیدا)^{۱۱}

اولین مرحله مهم در تشخیص سرقت ادبی با این روش، بازیابی یک زیرمجموعه از D_s که متشکل از موارد احتمالی سرقت ادبی در یک سند مظنون باشد. این روش شامل مقایسه گسترده سند مظنون با هر تعداد از D_s ها برای شناسایی شباهت‌های محلی می‌باشد. معمولاً تعداد سندهای منبع برای یک سند مظنون معین از قبل مشخص نیست.

تشخیص مرز متنی

هنگامی که مجموعه‌ای از اسناد منبع که با یک سند مظنون منطبق هستند، تعیین می‌شوند، مرحله بعدی عبارتست از انجام یک آنالیز دقیق برای ارزیابی مرزهای^{۱۲} دقیق متن‌های سرقت شده در هر دو سند منبع و مظنون. فرض کنیم که $D_{tx} \subseteq D_s$ نشان‌دهنده مجموعه اسناد منبع باشد که برای سند مظنون d_x بازیابی شده‌اند. هدف این روش یافتن SWNG‌هایی در پروفایل‌های d_x و هر $d_s \in D_{tx}$ و ایجاد جملات ماکسیمال از آنها است که با متن متناظر هستند.

در تصویر ۲، الگوریتم مربوط به تشخیص مرزهای متنی نشان داده شده است.

^{۱۱} Candidate text retrieval
^{۱۲} Boundaries

Input: dx, a suspicious document
 ds, a source document
 n2, length of stopword sequences
 θg , threshold of maximum gap allowed
Output: a set of detections

```

detectPassageBoundaries(dx, ds, n2,  $\theta g$ )
1. Px=profile(n2, dx);
2. Ps=profile(n2, ds);
3. [M1, M2]=match(Px, Ps);
4. InitPlagPass=findPassages(M1,  $\theta g$ );
5. Detections=[];
6. PlagPassages=[];
7. for all Pi  $\in$  InitPlagPass
8.   Oi=subset(Pi, M2);
9.   OrigPassages=findPassages(Oi,  $\theta g$ );
10.  if size(OrigPassages)>1
11.    for all Oj  $\in$  OrigPassages
12.      Pj=subset(Oj, M1)
13.      PlagPassages=PlagPassages  $\cup$  Pj;
14.    endfor
15.  else PlagPassages=PlagPassages  $\cup$  Pi;
16.  endif
17. Detections=Detections  $\cup$ 
    [PlagPassages, OrigPassages];
18. endfor
19. return Detections;

```

تصویر ۲: الگوریتم مربوط به تشخیص مرزهای متنی

فراپردازش^{۱۳}

روشی که مطرح شد مبتنی بر نمایش SWNG است و همه کلماتی که متعلق به مجموعه ۵۰ کلمه کلیدی

انتخابی نباشند را حذف می‌کند. تشخیص‌های انجام شده باید بررسی شوند تا مشخص شود که شباهت

میان متن سرقت شده شناسایی شده و متن اصلی، در هنگامی که کل متن مدنظر قرار گیرد بالا است. به

^{۱۳} Post-Processing

علاوه ما به مکانیسمی نیاز داریم که موارد سرقت شناسایی شده را سه درجه شباهت با متن اصلی نمره دهی کند.

روش تجزیه مقدار منفرد^{۱۴}

تجزیه مقدار منفرد (SVD) در حقیقت یکی از مهم‌ترین ابزارها در بازیابی اطلاعات به صورت «شاخص گذاری معنای نهانی» است. همچنین این روش یکی از مناسب‌ترین ابزارها از نظر به کارگیری ماتریس‌های پراکنده می‌باشد.

فرض: فرض کنیم که A یک ماتریس r با رتبه $m \times n$ باشد. همچنین فرض کنیم که $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$.
 مقادیر آنگن ماتریس $\sqrt{AA^T}$ باشند. سپس ماتریس‌های قطری $U = (u_1, \dots, u_r)$ و (v_1, \dots, v_r)
 $V = (v_1, \dots, v_r)$ که بردارهای ستونی آن‌ها قطری است و ماتریس $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$
 وجود دارند. تجزیه $A = U \Sigma V^T$ را تجزیه مقدار منفرد ماتریس A می‌نامند و اعداد $\sigma_1, \dots, \sigma_r$
 مقادیر منفرد ماتریس A هستند. ستون‌های U (یا V)، بردارهای چپ (یا راست) منفرد ماتریس A
 نامیده می‌شوند.

در اینجا ما تجزیه ماتریس اصلی A را داریم. نیازی به گفتن نیست که بردارهای منفرد چپ و راست، پراکنده نیستند. ما حداکثر r منفرد غیر صفر را در اختیار داریم که رتبه r بعد کوچکتر در میان دو بعد ماتریس است. با این حال ما فضای زیادی را در حافظه از طریق ذخیره‌سازی ماتریس عبارت-در-سند بدین شکل اشغال می‌کنیم. خوشبختانه از آنجا که مقادیر منفرد معمولاً به سرعت پایین می‌آیند، ما می‌توانیم تنها

^{۱۴} Singular Value Decomposition

k مورد از بزرگ‌ترین مقادیر منفرد را به همراه مختصات برداری منفرد متناظر آن‌ها انتخاب کنیم و یک

تجزیه منفرد کاهش یافته به k از A را بوجود آوریم.

فرض کنیم که k ، $0 < k < r$ و تجزیه مقدار منفرد A را داریم

$$A = U \Sigma V^T = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

ما $A_k = U_k \Sigma_k V_k^T$ را تجزیه مقدار منفرد کاهش یافته به k (rank- k SVD) می‌نامیم.

در بازیابی اطلاعات اگر هر سند مرتبط با یک موضوع باشد که ما می‌توانیم معانی نهان (latent semantics) را

به دست آوریم- کلمات و اسناد مرتبط از نظر معنایی دارای بردارهای یکسانی در فضای کاهش یافته

خواهند بود. برای نمایش تجزیه rank- k SVD، تصویر ۳ را مشاهده کنید. نواحی خاکستری تعیین‌کننده

اولین k مورد از مختصات بردارهای منفرد هستند که مورد استفاده قرار می‌گیرند.

$$\begin{bmatrix} A_k \end{bmatrix}_{m \times n} = \begin{bmatrix} U_k \end{bmatrix}_{m \times k} \begin{bmatrix} \Sigma_k \end{bmatrix}_{k \times k} \begin{bmatrix} V_k^T \end{bmatrix}_{k \times n}$$

تصویر ۳: تجزیه مقدار منفرد کاهش یافته به k

نقشه های خود-آرایی (Kohonen) [۱]

یک نقشه Kohonen را همچنین نقشه خود-آرایی^{۱۵} (SOM) نیز می نامند که در سال ۱۹۸۹ توسط

کوهوننیس به وجود آمد. این یک ابزار شبکه عصبی مصنوعی رقابتی است. SOM یکی از مؤثرترین ابزارها برای

مشاهده بصری داده های چند بعدی تلقی می شود و همچنین مکانیسم مؤثری در پردازش سیگنال و

کارکردهای استخراج داده ای است. SOM گونه ویژه ای از شبکه های عصبی است که می تواند یک سیگنال

ورودی چند بعدی پیچیده را به یک سیگنال کم-بعدتر نگاشت یا ساده سازی کند. از آن برای دسته بندی

مجموعه داده ای با توجه به شباهت های مجموعه استفاده می شود. نقشه خود-آرایی یک شبکه مصنوعی

است که در دو لایه نورون سازماندهی می شود. اولین لایه نشان دهنده داده های ورودی و دومین لایه

نشان دهنده یک گرید نورون هستند، که اغلب دوبعدی بوده و کاملاً یکدیگر مرتبط می باشند. همه گره های

ورودی به همه نورون های (گره های) خروجی متصل می باشند. نورون های خروجی معمولاً در دو یا سه

گرید کم-بعد آرایش می یابند. یک بردار وزنی برای هر نورون وجود دارد که دارای بعدیتی همانند بردارهای

ورودی است. تعداد بعد های گرید خروجی معمولاً پایین تر از ابعاد ورودی است. SOM ها اساساً برای کاهش

بعدیت به کار گرفته می شوند و نه برای انبساط.

ماتریس فاصله یکپارچه

ماتریس فاصله یکپارچه^{۱۶} یا ماتریس U که نمایشی است از یک نقشه خود-آرایی که فواصل میان نورون

های شبکه و یا ماتریس های واحد را نشان می دهد، ابزار مؤثری برای مشاهده دسته ها در داده های ورودی

^{۱۵} Self Organization Map
^{۱۶} Unified Distance Matrix

بدون وجود هرگونه اطلاعات ما قبل در مورد دسته‌ها است. نمایش چند بعدی داده‌های نقشه‌ای خود-آرایی

می‌تواند با استفاده از ماتریس فاصله یکپارچه انجام شود. این مسئله با استفاده از روابط توپولوژیکی میان

نورون‌ها پس از فرایند یادگیری، حاصل می‌آید. ماتریس U دارای فواصل از هر مرکز واحد به همه

همسایه‌های خود است. نورون‌های شبکه SOM در اینجا از طریق سلول‌های شش ضلعی نشان داده

می‌شوند. با استفاده از ماتریس U ما می‌توانیم روابط توپولوژیکی میان نورون‌ها را شناسایی کنیم و در مورد

ساختار داده ورودی استنباط‌هایی را انجام دهیم. از الگوهای رنگی برای نمایش شباهت‌ها استفاده می‌شود.

رنگ‌های تیره (مقادیر بالا در ماتریس U) نشان دهنده آن هستند که فاصله‌ای میان مقادیر در فضای ورودی

وجود دارد و مشخص می‌کند که هیچ شباهتی وجود ندارد. همچنین رنگ‌های روشن (مقادیر پایین در

ماتریس U) نشان می‌دهند که بردارهای در فضای ورودی به هم نزدیک هستند و این بیان‌کننده

شباهت‌های زیاد در میان ورودی‌ها است. از این طریقه نمایش می‌توان برای بررسی ساختار فضای ورودی و

دریافت دیدگاهی از ساختار ورودی غیر مشهود در فضای داده چند بعدی استفاده نمود.

مزایا و معایب: مهمترین مزایایی که این روش بدنبال آنها است عبارتند کاهش زمان جستجو در تشخیص

سرقت ادبی و ساده سازی نمایش بصری و آنالیز نتایج.

برچسب زنی نقش معنایی [۴]

برچسب زنی نقش معنایی^{۱۷} (SRL) یکی از روش‌های پردازش زبان طبیعی است که در حوزه‌های بسیاری

مانند خلاصه سازی متن، دسته‌بندی متن و طبقه‌بندی متن کاربرد دارد. شیوه تشخیص سرقت ادبی بر

^{۱۷} Semantic Role Labeling

مبنای SRL می‌تواند انواع سرقت ادبی شامل copy-paste، تغییر کلمات، جایگزینی مترادف‌ها، تغییر ساختار

کلمات در جمله، ترکیب جملات از حالت غیر فعال به فعال و بالعکس را شناسایی کند. SRL برای ارزیابی

معنای جملات به کار گرفته می‌شود و واژه نامه WordNet نیز برای استخراج مفاهیم یا مترادف‌ها برای هر

کلمه در داخل جملات مورد استفاده قرار می‌گیرد. نمره وزنی برای هر آرگومان در جهت بررسی رفتار و

تأثیرات آن‌ها در تشخیص سرقت ادبی محاسبه می‌شود.

در این روش ابتدا اسناد مظنون و اسناد اصلی با استفاده از تفکیک متن، حذف کلمات کلیدی، و ریشه‌یابی

پیش پردازش می‌شوند. سپس از SRL برای تبدیل جملات به آرگومان‌ها بر مبنای موقعیت هر عبارت در

جمله استفاده می‌شود. فعل‌های جملات نقش مهمی را در فرایند ایفا می‌کنند و آنالیز جملات مبتنی بر

فعل‌های آن‌ها است. همه آرگومان‌های استخراج شده از متن، براساس نوع آرگومان در گروه‌هایی گروه بندی

می‌شوند. هر گروه دارای آرگومان‌های استخراج شده مشابهی است و به واسطه نام آرگومان، نام‌گذاری

می‌شود، مانند Arg⁰، Arg¹، V، Time، Location... این مرحله را «گروه بندی برچسب آرگومان» (ALG)

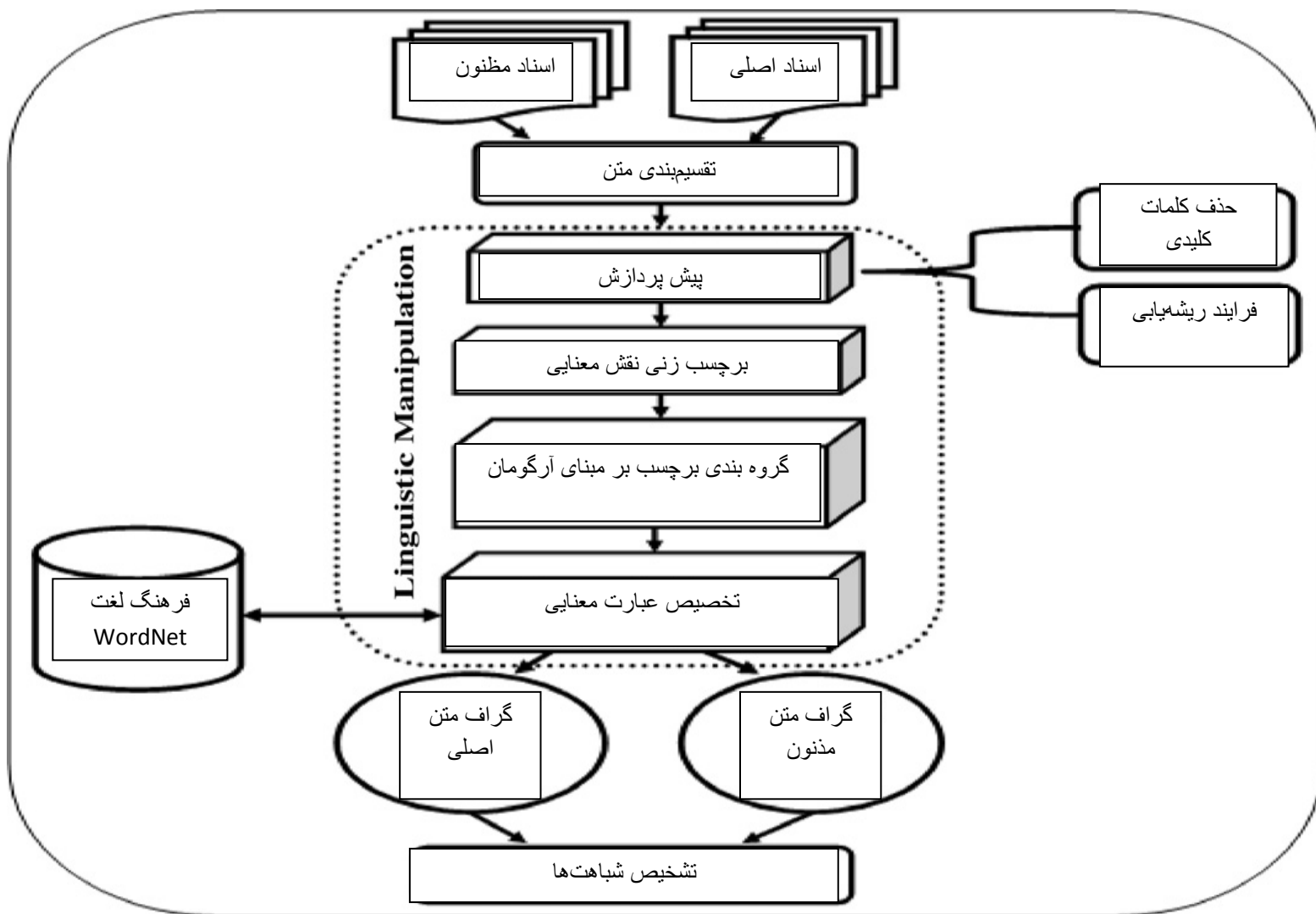
می‌نامند. سپس ما همه مفاهیم هر عبارت را در گروه‌های آرگومان با استفاده از فرهنگ لغت WordNet

استخراج می‌کنیم. این مرحله را تخصیص عبارت معنایی (STA) می‌نامیم. همه مفاهیمی که توسط فرهنگ

لغت WordNet استخراج شدند، در یک گروه با نام «گروه موضوعی» گردآوری می‌شوند. مزیت یک گروه

موضوعی آن است که به سرعت می‌توانند بخش‌های مظنون را در اسناد پیدا کند. تصویر ۴ نشان دهنده

ساختار عمومی برای این روش است.



تصویر ۴: ساختار روش SRL

درهم سازی جملات [۵]

مزیت اصلی الگوریتم درهم سازی جملات^{۱۸}، استفاده از روش درهم سازی برای جملات سند پردازش شونده است.

ابتدا هر سند به صورت تعدادی از جملات مجزا شده توسط یک نقطه مشاهده می‌شود. در حین پردازش متن، مکانیسم‌هایی وجود دارد که برای شناسایی مترادف و یا نقاط اعشار در اعداد به کار گرفته می‌شوند. این

^{۱۸} Sentence Hashing

مسئله از آنجا اهمیت دارد که بسیاری از روش‌ها با نرمال سازی متن آغاز می‌شوند. عملکرد این الگوریتم به‌طور کلی تا حدی می‌تواند تحت تأثیر عدم نرمال سازی داده‌های ورودی قرار گیرد، و ذخیره سازی ساختار جمله برای موفقیت آن ضروری است. مسلماً متن می‌تواند دربرگیرنده تعدادی از بخش‌های ساختار بندی نشده مانند فهرست‌ها و لیست‌ها باشد. بدین ترتیب ایجاد فریم می‌تواند شرایطی که در آن نقطه ایدر پایان جملات وجود ندارد را مدیریت کند.

سپس، طولی از فریم درهم برای کل سیستم انتخاب می‌شود. فریم درهم از تعدادی از عبارات پی‌درپی حاصل از سند پردازش شده پدید می‌آید. طول این فریم بسیار اهمیت دارد زیرا در ارتباط مستقیم با عملکرد الگوریتم است. این فریم هدایت کننده ایجاد درهم‌های (hash) متعاقب می‌باشد. این فریم نباید بیش از حد کوتاه باشد زیرا هزینه‌های دفتری از جنبه‌های مثبت درهم سازی در مراحل بعدی الگوریتم، فراتر می‌روند؛ و همچنین نباید بیش از حد بلند باشد زیرا بازدهی جستجو برای بلندترین چشمگیری کاهش پیدا می‌کند.

روش درهم سازی به واسطه دو پارامتر هدایت می‌شود: α و β . در هنگام پردازش متن، دنباله‌ای از عبارات به صورت یک فریم متنی منفرد در نظر گرفته می‌شود که باید در هنگام که طول کلی آن کمتر از مقدار پارامتر α باشد، در هم سازی شود، در غیر این صورت دنباله به تعدادی از فریم‌های متنی تقسیم می‌شود که طول آنها β است. خواه نتایج تخصیص دارای باقی مانده باشد یا خیر (که طول آن کمتر از مقدار α پس) آن را بر مبنای بخش اول قاعده پردازش می‌کنیم.

هر عبارت در فریم در یک عبارت منحصر به فرد نگاشت می‌شود، سپس مقدار مربوط به یک فریم معین به صورت مجموعه اعداد نشان دهنده آن فریم معین، محاسبه می‌شود. بدین ترتیب، این مسئله باعث می‌شود که روش مورد نظر در مقابل ترتیب کلمات در عبارات مورد بررسی انعطاف‌پذیری داشته باشد، که یکی از ویژگی‌های بسیار مهم در کارکردهای تشخیص سرقت ادبی است. کل متن پردازش می‌شود تا جمله‌ای با اعداد نمایانگر یک سند یافت شود.

مثال زیر را با پارامترهای α برابر با ۹ و β برابر با ۱,۵ در نظر بگیرید. این مقادیر باید به گونه‌ای تفسیر شوند که جمله‌ای تا ۱۳ کلمه به صورت یک فریم متنی منفرد در نظر گرفته شود ($\alpha * \beta$) و جملات بزرگ‌تر از آن به فریم‌هایی ۹-عبارتی افزای گردند.

On Tuesday Huntsman Corporation informed that it had revoked a bid worth \$460 million for Rexene Corp. due to double rejection of its bids by the chemical company from Dallas.

On → 742 Tuesday → 110226 Huntsman → 289972 Corporation → 88818 informed → 57582 that → 66940 it → 956 had → 7532 revoked → 81758 a → 97

Frame hash: 704623

bid → 7212 worth → 161187 \$460 → 22240 million → 33767 for → 7530 Rexene → 38381 Corp → 42432 due → 7437 to → 1039 double → 276024

Frame hash: 597249

rejection → 169746 of → 990 its → 7763 bids → 57811 by → 905 the → 8357 chemical → 337263 company → 354349 from → 60517 Dallas → 340253

Frame hash: 1337954

جدول ۲: مثال

مزایا و معایب: در روش درهم سازی، طول جملات یک مانع محسوب می‌شود. مهمترین کاری که می‌توان در این مورد انجام داد عبارت است از متراکم سازی معنایی برای کاهش بیشتر بردار مطرح شده.

مدل و نمونه [۶]

نکته کلیدی در این روش چگونگی اندازه‌گیری شباهت‌هاست. الگوریتم‌های قدیمی تر سرقت ادبی را از طریق انطباق رشته به واسطه مقایسه متن دو سند مشخص می‌کنند. برخی از ابزارها مبتنی بر این‌گونه الگوریتم‌ها هستند. محدودیت آن‌ها آنست که تنها می‌توانند متونی را بیابند که دقیقاً یکسان هستند و هیچ گونه تغییراتی در متن را نمی‌توانند کنترل کنند. یک روش بهینه عبارت است از محاسبه تناوب کلمات در سند و نمایش سند به صورت یک بردار مشخصه (ویژگی)، و سپس محاسبه شباهت از طریق معیار شباهت کسینوسی. این روش می‌تواند تشخیص سرقت ادبی در هنگام جابجا شدن لغات یا جملات و یا حذف یا افزوده شدن برخی کلمات انجام دهد. این گونه روش‌ها می‌توانند سرقت ادبی را با دقت بالاتری پیدا کنند. برخی از ابزارهای عملی با این روش‌ها ساخته می‌شوند.

در مقاله‌ای توسط ژانگ و فانگ، معیار شباهت کسینوسی^{۱۹} از دو جنبه بهبود پیدا کرده است. جنبه اول پردازش سند است. پیش پردازش از آن جا لازم است که در یک سند ممکن است کلمات جابه‌جا شده باشند و یا کلماتی با مفاهیم مشابه و یا کلمات کمکی استفاده شده باشند. پس از پیش پردازش، کلماتی با مفاهیم مشابه گروه بندی می‌شوند و کلمات کمکی از بردار مشخصه حذف می‌شوند. بهسازی دوم عبارت است از تنظیم وزن کلمات. اگر کلمه A در نوعی از اسناد مشابه کتابخانه به کار گرفته شده باشد، این به معنای سرقت ادبی نیست حتی اگر کلمه در هر دو سند چندین بار استفاده شده باشد. درغیراین‌صورت، اگر کلمه B با تناوب پایین در کتابخانه وجود داشته باشد، و در هر دو سند به طور متناوب استفاده شده باشد،

^{۱۹} Cosine Similarity Measure

سپس احتمال سرقت ادبی بسیار بالاتر است. بدین ترتیب، در بردار مشخصه، وزن کلمه A باید کاهش پیدا کند و وزن کلمه B باید افزایش یابد.

مزایا و معایب: عدم پیش-پردازش صحیح سند باعث ایجاد خطا در نتایج می شود و وزن کلمات سند نیز باید مبتنی بر تناوب تکرار آنها در کتابخانه دیجیتال باشد زیرا در غیر این صورت شاخص شباهت کسینوس نمی تواند دقت کافی داشته باشد.

متن تکراری [۷]

جملات دارای کلمات مشابه بین سندهای مظنون و منبع می توانند اولین نشانه سرقت ادبی به شمار روند. با این حال وجود آن ها به عنوان یک نشانه منحصر به فرد از سرقت ادبی نمی تواند چندان مورد اطمینان و اعتماد باشد، زیرا کاربری موضوعی کلمات نیز می تواند منجر به ایجاد جملاتی با کلمات مشابه (مثبت های کاذب) شود. به علاوه حتی تغییرات کوچکی برای مخدوش ساختن سرقت ادبی مانع از شناسایی جملات متناظر شده و منفی های کاذب پدید می آورد.

به منظور کنترل مشکلات بالا، سانچز وگا و همکاران یک استراتژی جدید را برای شناسایی یک متن سرقت شده با نام روش «شاخص بازنویسی» ارائه کرده است. این روش می تواند بخش های متن تکراری را حتی در صورت ایجاد تغییرات در آن تشخیص دهد. به علاوه، هدف آنها تسهیل تشخیص سرقت ادبی به واسطه بررسی ویژگی های متنوع بخش های مربوط به متن تکراری در طی مرحله دسته بندی می باشد. روش شاخص بازنویسی می تواند یک وزن را به هر کلمه مربوط به سند مظنون اختصاص دهد تا درجه عضویت آن ها در

بخش متن سرقت شده نشان دهد. بدین ترتیب امکان کشف متنی که تغییراتی در آن اعمال شده است (مانند حذف لغات، جایگذاری لغات، و تغییر لغات) وجود دارد و این مسئله امکان یک انطباق جزئی را میان اسناد فراهم می‌آورد.

یک موضوع مهم دیگر که مورد توجه این محققین قرار گرفت، نمایش جامع‌تر بخش‌های متن تکراری^{۲۰} است. این گونه نمایش به الگوریتم‌های دسته‌بندی امکان می‌دهد که تمایز بیشتری را میان اسناد سرقت شده و سرقت نشده از طریق ارائه ویژگی‌هایی که توصیف کننده تعداد بخش‌های متن تکراری و نیز اهمیت و دسته‌بندی آن‌ها می‌باشد، قائل شوند.

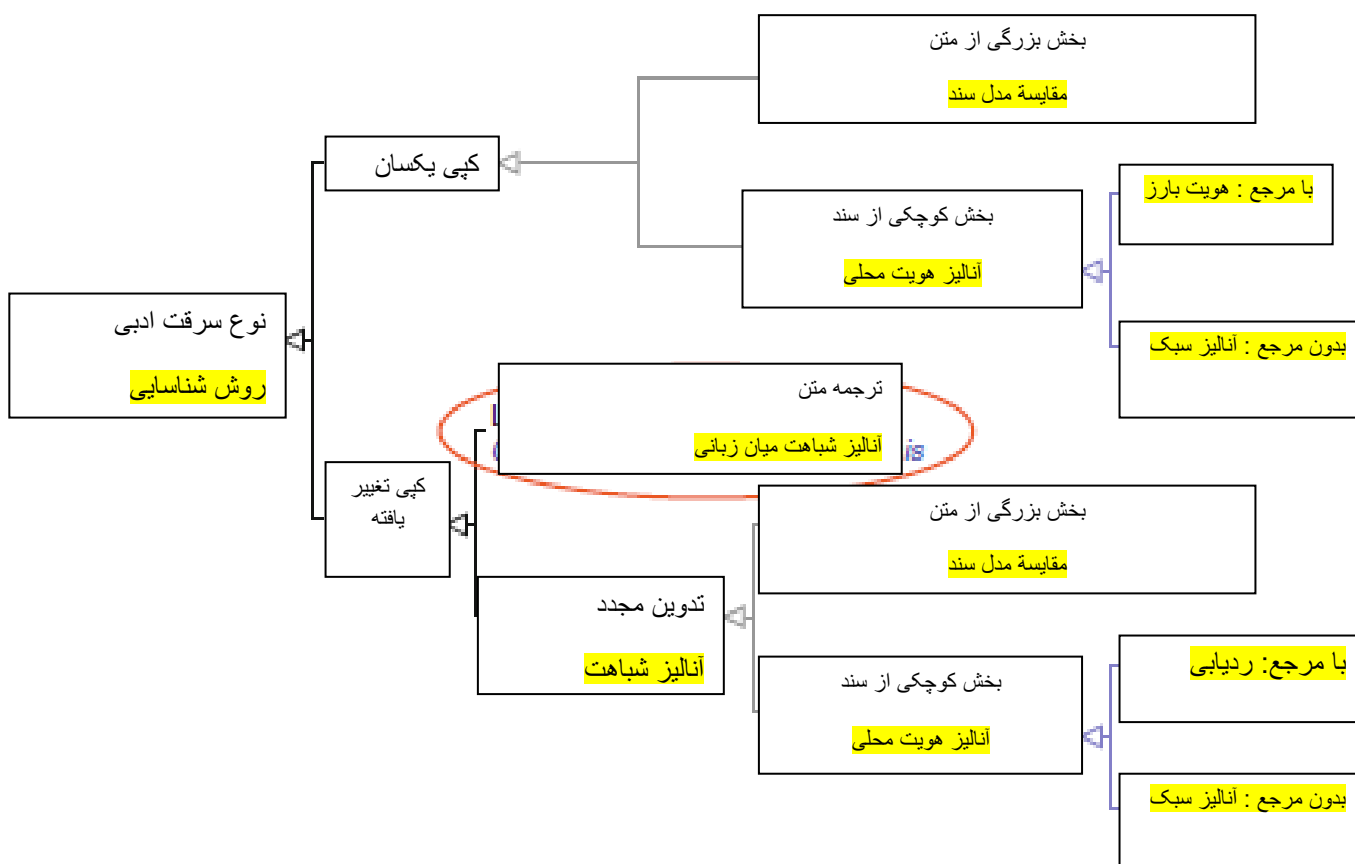
مزایا و معایب: این روش می‌تواند بخش‌های متن تکراری را حتی در صورت قرار گرفتن تحت تغییرات مختلف شناسایی کند. همچنین در این روش تشخیص سرقت ادبی از طریق مد نظر قرار دادن ویژگی‌های مختلف بخش‌های تکراری متن در طی مرحله دسته‌بندی، تسهیل می‌شود.

فصل سوم : سرقت ادبی

میان-زبانی

سرقت ادبی میان زبانی در حقیقت مرتبط است با شناسایی اتوماتیک و استخراج سرقت ادبی در محیط‌های

چند-زبانی. یک طبقه‌بندی از انواع سرقت ادبی و نیز روش‌های شناسایی آن‌ها در تصویر ۵ ارائه شده است.



تصویر ۵: طبقه بندی انواع سرقت ادبی متنی، به همراه روش‌های شناسایی آن‌ها [۸]

انواع روش‌های شناسایی سرقت ادبی میان زبانی

سیستم‌های مبتنی بر لغت

این سیستم‌ها مبتنی هستند بر شباهت‌های لغات^{۲۱} میان زبان‌ها (مثلاً انگلیسی-فرانسه) و تأثیرات زبانی

(مثلاً computer در انگلیسی -> computadora در اسپانیایی) میان زبان‌ها. شباهت‌های لغات در زمان‌های

مختلف می‌تواند در شکل‌گیری عبارات کوتاه منعکس شود؛ مثلاً پیشوند ها و یا n گرم های حرفی (کاراکتر).

^{۲۱} Lexicon

شاید دو مورد از اولین مدل‌های شباهت از این دست را بتوان «هم ریشه» بودن (بر اساس پیشوندها و سایر نشانه‌ها) و نمودار نقطه ای (بر مبنای ۴-گرم‌های کاراکتر) دانست. درحالی‌که این مدل‌ها اساساً برای هم‌تراز ساختن متون دوگانه پیشنهاد می‌شوند، اما آن‌ها برای تشخیص کاربری موجود تکراری در میان زبان‌ها (با برخی محدودیت‌ها) نیز کاربرد دارند.

سیستم‌های مبتنی بر فرهنگ لغت

این سیستم‌ها لغات یا مفاهیمی مانند نام ماهیت‌ها را در یک فضای نمایش مشترک از طریق یک فرهنگ لغت^{۲۲} چند زبانی نگاشت می‌کنند. با این حال فرهنگ‌های لغت چند زبانی همیشه در دسترس نیستند و همچنین Ceska دریافت که ناقص بودن فرهنگ لغت می‌تواند قابلیت‌های تشخیص را محدود کند.

سیستم‌های مبتنی بر مجموعه‌های داده قابل مقایسه

این سیستم‌ها با استفاده از مجموعه‌های داده قابل مقایسه^{۲۳} آموزش می‌بینند. یک نمونه از این امر را می‌توان آنالیز معنایی صریح میان زبانی دانست (CL-ESA). d_q و d' از طریق بردار شباهت‌ها با اسناد یک مجموعه شاخص CL، C_l ، نمایش داده می‌شوند، یعنی

$$\vec{d_q} = \{sim(d_q, c_1), \dots sim(d_q, c_l)\}, \vec{d'} = \{sim(d', c'_1), \dots sim(d', c'_l)\} (c_l \in L, c'_l \in L')$$

که

در آن sim یک مدل شباهت درون زبانی است، مانند معیار کسینوسی، و $\vec{d_q}$ و $\vec{d'}$ برای محاسبه $sim(d_q, d')$ مورد مقایسه قرار می‌گیرند.

^{۲۲} Thesaurus
^{۲۳} Comparable Corpus

سیستم‌های مبتنی بر مجموعه‌های داده موازی

این سیستم‌ها با استفاده از مجموعه‌های داده ای موازی^{۲۴}، برای یافتن تشابه‌های میان زبانی و یا یافتن مدل‌های ترجمه آموزش داده می‌شوند. اصول و منابع ماشین ترجمه (MT) در آن‌ها به کار گرفته می‌شود اما هیچ ترجمه حقیقی انجام نمی‌شود.

سیستم‌های مبتنی بر ترجمه ماشینی [۹]

این مدل‌ها در CLPD رایج هستند و وظیفه مورد نظر را با تبدیل آن به یک مسئله تک-زبانی آسان‌تر می‌کنند. مسئله نمونه به صورت زیر است: ۱. یک شناساگر زبانی برای تعیین محتمل‌ترین زبان d_q به کار گرفته می‌شود؛ ۲. d_q در صورتی که به زبان مقایسه نوشته نشده باشد، ترجمه می‌شود؛ ۳. یک مقایسه درون زبانی بین d_q و d' انجام می‌شود.

توصیف برخی از روش‌های مورد استفاده برای تشخیص سرقت ادبی میان زبانی

Ferret [۱۰]

Ferret ابزاری است برای شناسایی موجود مشابه متنی در مجموعه‌های بزرگی از اسناد. از آن با موفقیت در متون انگلیسی برای سال‌ها استفاده شده است. این یک ابزار رایگان و مستقل است که برای کاربران مبتدی و برای اجرا روی کامپیوترهای معمولی طراحی شده است و نتایج متوسطی را به دست می‌دهد. این ابزار دربرگیرنده تعداد زیادی از اسناد است، مانند مقالاتی که توسط تعداد بسیاری از دانشجویان ارائه شده‌اند. همچنین از آن می‌توان برای شناسایی سرقت ادبی در کدهای کامپیوتری نیز استفاده کرد. بائو و همکارانش

^{۲۴} Parallel Corpus

یک نسخه اصلاح شده از این ابزار را به کار گرفتند و مشاهده کردند که عملکرد آن در مورد سرقت ادبی در

متون زبان چینی جالب توجه است. آثار دانشجویی از دو دانشگاه چین گردآور شدند و از Ferret برای

تشخیص سرقت زدگی استفاده شد. نتایج به دست آمده توسط این محقق مشخص می کنند که Ferret می تواند

سرقت ادبی مصنوعی و حقیقی (که پیش تر کشف نشده است) را شناسایی کند.

یک سیستم جالب توجه دیگر برای تشخیص کپی Turnitin است که از پایگاه داده بسیار بزرگی از مطالب

روی وب و آثار پیشین دانشجویان استفاده می کند و آن ها را با آثار کنونی مقایسه می کند. با این حال اسناد

باید برای پردازش در اختیار Turnitin قرار گیرند ، و یک هزینه تجاری برای آن وجود دارد. مقایسه ای از

Ferret، Turnitin و سایر سیستم ها توسط Lyon (۲۰۰۳) ارائه شده است. روش های دیگری وجود دارند که به

شباهت های معنایی میان زوج های اسناد توجه می کنند (بائو ، ۲۰۰۴ و ۲۰۰۶). تشخیص کپی برداری در

کدهای کامپیوتری توسط این ابزار نیز موضوع بررسی های مالفول (۲۰۰۶) بود.

طرح سیستم Ferret

شناساگر کپی Ferret مجموعه ای از فایل ها را در اختیار می گیرد و شاخصی از شباهت را برای هر زوج از

آن ها محاسبه می کند. مرحله اول، تبدیل یک سند به مجموعه ای از تری گرم های متداخل است. بدین

ترتیب جمله ای مانند:

A storm is forecast for the morning

به مجموعه تری گرم های زیر تبدیل می شود:

A storm is storm is forecast is forecast for

forcast for the for the morning

سپس مجموعه تری گرم ها برای هر سند با همه موارد دیگر مقایسه می شود و شاخص شباهت هر زوج از

اسناد محاسبه می گردد. معمولاً نتایج به شکل یک جدول رتبه بندی شده ارائه می شوند که در آن مشابه

ترین زوج ها در بالا قرار دارند.

سه استراتژی زیر برای پردازش رشته های کاراکتر هایی که یک جمله چینی را تشکیل می دهند برای انطباق

Ferret با زبان چینی مورد استفاده قرار گرفتند:

استراتژی ابتدایی: کاراکترهای چینی همانند کاراکترهای انگلیسی مورد نظر قرار گرفتند؛ دنباله ها با مدنظر

قرار دادن هر عنصری که کاراکتر چینی نیست (مانند فاصله، علایم نگارشی، اعداد و غیره) به عنوان یک رمز

نشانه ای از یکدیگر تفکیک شدند .

استراتژی کاراکتر منفرد: به جای یافتن کلمات، کاراکترها به طور منفرد پردازش می شدند. هر کاراکتر منفرد

در فایل متنی عنوان یک نشانه به کار گرفته می شود.

استراتژی فرهنگ لغت: بر مبنای یک فرهنگ لغت چینی، یک جمله به دنباله کلمات شناسایی شده در

فرهنگ لغت تفکیک می شود.

N-گرم ها [۱۱]

مدل n-گرم ابتدا در دسته‌بندی متون بر مبنای اطلاعات آماری گرفته‌شده از کاربری دنباله کاراکترها مورد

استفاده قرار گرفت. n-گرم ها در حقیقت کاراکترهای متداخل تناوبی شکل گرفته از یک جریان ورودی

می‌باشند. آن‌ها را می‌توان عنوان جایگزینی برای بازیابی کلمه-محور متن مورد استفاده قرار داد.

محمد آ. خان و همکارانش یک سیستم تشخیص سرقت ادبی را برای موجود زبان اردو بر مبنای مدل n-

گرم پیشنهاد کردند. آن‌ها از تری گرم برای نمایش متن بهره گرفتند. تری گرم بدان معناست که نشانه سه

کلمه برای استخراج کلمات از متن مورد استفاده قرار می‌گیرد و این تری گرم ها با هم منطبق می‌شوند.

سپس معیارهای انطباقی برای دسته‌بندی متن محاسبه می‌گردند. معیار شباهت R به‌صورت زیر تعریف

می‌شود:

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}$$

در زیر دو متن ۱ و ۲ مشاهده می‌شوند. متن ۱ در حقیقت متن اصلی (گرفته شده از یک کتاب به زبان

اردو) است و متن ۲ نسخه دیگری از متن ۱ با تغییر کلمات است. تری گرم های مربوط به هر دو متن

محاسبه می‌شوند؛ جدول ۲ فهرستی از تری گرم های محاسبه شده برای متن ۱ و جدول ۳ فهرستی از تری

گرم های محاسبه شده برای متن ۲ را ارائه می‌کنند.

متن ۱:

برسات کا موسم شروع ہو چکا تھا۔ آسمان پر ہر وقت کالی کالی بدلیاں اٹکھیلیاں کرتی نظر آتیں اور ہلکی ہلکی بوندا باندی موسم کی خوبصورتی میں اضافہ کر دیتی۔ مگر ایک مسئلہ تھا۔

متن ۲:

برسات کا موسم جب شروع ہوتا ہے تو آسمان پر ہر طرف کالے بادل نظر آتے ہیں۔ اور ہلکی ہلکی پوہار موسم کو خوشگوار بنا دیتی ہے۔ مگر ایک مسئلہ ہے۔

برسات کا موسم	کا موسم شروع	موسم شروع ہو	شروع ہو چک	ہو چکا تھا	چکا تھا آسمان
تھا آسمان پر	آسمان پر ہر	پر ہر وقت	ہر وقت کالی	وقت کالی	کالی کالی بدلیاں
کالی بدلیاں	بدلیاں اٹکھیلیاں	اٹکھیلیاں کرتی	کرتی نظر	نظر آتیں اور	آتیں اور ہلکی
اور ہلکی ہلکی	ہلکی ہلکی بوندا	ہلکی بوندا باندی	بوندا باندی	باندی موسم	موسم کی
کی خوبصورتی	خوبصورتی میں	میں اضافہ کر	اضافہ کر	کر دیتی مگر	دیتی مگر ایک
مگر ایک مسئلہ	ایک مسئلہ تھا				

جدول ۳: فہرست تری-گرم ہا برای متن ۱

برسات کا موسم	کا موسم جب	موسم جب شروع	جب شروع ہوتا	شروع ہوتا ہے	ہوتا ہے تو
ہے تو آسمان	تو آسمان پر	آسمان پر ہر	پر ہر طرف	ہر طرف کالے	طرف کالے بادل
کالے بادل نظر	بادل نظر آتے	نظر آتے ہیں	آتے ہیں اور	ہیں اور ہلکی	اور ہلکی ہلکی
ہلکی ہلکی پوہار	ہلکی پوہار موسم	پوہار موسم کو	موسم کو	کو خوشگوار بنا	خوشگوار بنا
بنا دیتی ہے	دیتی ہے مگر	ہے مگر ایک	مگر ایک مسئلہ	ایک مسئلہ ہے	

جدول ۴: فہرست تری-گرم ہا برای متن ۲

Passage	N	M	R	% Matching	Comments
J1	32	4	0.07	7%	Different
J2	29				
B1	48	26	0.3	30%	No Same
B2	44				
C1	64	6	0.04	4%	No Different
C2	74				
D1	42	10	0.133	13.3%	No
D2	43				
E1	40	19	0.3	30%	No Same
E2	45				
F1	44	37	0.73	73%	Yes Copied
F2	41				
G1	41	10	0.147	14.7%	No
G2	37				
H1	47	7	0.08	8%	Different
H2	38				
I 1	48	2	0.02	2%	Different
I 2	30				
A1	90	90	1	100%	Same
A2	90				

جدول ۵: نتایج مفید توصیف شده. ستون Passage شامل متن ها است، N تعداد کلی تری-گرم ها در متن است، M تعداد تری-گرم های منطبق می باشد، و R ضریب تشابه است.

مزایا و معایب	روش
این روش یک ابزار رایگان و مستقل و البته بسیار سریع است که کاربران مبتدی نیز می توانند از آن استفاده کنند.	Ferret
تشخیص کپی توسط ۲-گرم ها ماکزیمم است اما پیچیدگی های استخراج و مقایسه آنها نیز ماکزیمم است. نرخ تشخیص کپی در مدل ۴-گرم کوچکتر از همه است و در مقایسه جملات طولانی، کوچکترین تعداد ۴-گرم های انطباقی را میابد. ۳-گرم ها عملکرد متوسط قابل قبولی را با هزینه مناسب از نظر پیچیدگی و آلارم خطا ارائه می دهد (با این حال بطور کلی n-گرم ها روشهای پر هزینه و زمانبری تلقی میشوند).	N-گرم ها

جدول ۶: مزایا و معایب روش های تشخیص سرقت ادبی میان - زبانی

فصل چهارم : مرور کوتاهی بر سرقت ادبی در صفحات وب

اینترنت هم اکنون یک فاکتور کلیدی در زندگی روزمره برای جمع‌آوری اطلاعات و استخراج داده‌ها و مفید از صفحات وب است. عملکرد و قابلیت اطمینان موتورهای جستجوی وب و دشواری‌های قابل توجهی به واسطه وجود مقادیر بسیار زیادی از داده‌های وب مواجه شده است. حجم بسیار زیاد اسناد کپی برداری شده و شبه کپی برداری شده منجر به ایجاد سرباره‌های مضاعف در موتورهای جستجو شده است، و کارکرد آن‌ها را شدیداً تحت تأثیر قرار داده است. نیاز به هم گذاری داده‌ها از منابع ناهمگن و مختلف، به ایجاد صفحات وب شبه کپی برداری شده انجامیده است. داده‌های شبه کپی برداری شده دارای شباهت زیادی با یکدیگر هستند اما در حقیقت «یکسان» نمی‌باشند. اسناد شبه کپی برداری شده در حقیقت صفحات وبی هستند که از نظر محتویات کمی با هم متفاوت‌اند. تفاوت میان این اسناد می‌تواند ناشی از عناصر گنجانده شده در آن‌ها باشد و نه خود محتویات اصلی صفحه. برای مثال، تبلیغات صفحات وب و یا برچسب‌های زمانی مربوط به زمان بروز رسانی یک صفحه وب، هر دو اطلاعاتی هستند که در هنگام جستجوی یک صفحه برای کاربر اهمیتی ندارند، و در نتیجه در هنگام مرور صفحات وب اطلاعات زیادی را به دست نمی‌دهند. وجود صفحات وب شبه کپی برداری شده ناشی از استفاده از مطالب سایت اصلی، سایت انعکاسی، سایت نسخه‌برداری شده، و نمایش‌های گوناگونی از یک آبجکت فیزیکی و اسناد سرقت شده می‌باشد. در بسیاری از شرایط مختلف، دو سندی که دقیقاً قابل تفکیک از یکدیگر نیستند ممکن است حاوی مطالب یکسانی باشند و آن‌ها را باید مطالب شده کپی برداری شده دانست. برای مثال، صفحات وب مربوط به سایت‌های انعکاسی مختلف ممکن

است تنها از نظر سربرگ و پانویست با هم متفاوت باشند. این گونه سندها از نظر مضامین سایت با هم فرقی ندارند بلکه در بخش‌های کوچک‌تر مانند تبلیغات و غیره متفاوت می‌باشند.

تشخیص اسناد شبه کپی برداری شده یک مسئله تحقیقاتی بسیار مهم در سال‌های اخیر بوده است. برنامه‌های کاربردی متعددی وجود دارند که می‌توانند به شناسایی اسناد شبه کپی برداری شده در حوزه تشخیص سرقت ادبی، تشخیص اسپم، و نیز در مرور متمرکز وب کمک کنند. در تشخیص سرقت ادبی، بخشی از یک سند که ممکن است یک جمله یا پاراگراف باشد، در سند دیگری گنجانده شده است و می‌توان آن دو سند را کپی یکدیگر دانست. پیام‌های اسپم متعلق به یک برنامه تبلیغاتی ممکن است بسیار متفاوت به نظر برسد زیرا اسپمرها اغلب باید پیام‌های جدیدی را منتقل کنند و عبارات یا پاراگرافهای نامرتبلی را برای عبور از فیلترها استفاده نمایند. با این حال این اسپم‌ها به راحتی از طریق روش‌های تشخیص اسناد شبه کپی برداری شده قابل کشف هستند. تعیین صفحات وب شما به کپی برداری شده به مرور متمرکز وب کمک می‌کند و از کیفیت و تنوع بالاتر نتایج جستجو اطمینان حاصل می‌نماید.

تحقیقات انجام شده

روش مربوط به ارزیابی درجه شباهت میان زوج‌های اسناد با عنوان شینگلینگ شناخته می‌شود. برودر و همکارانش روشی را برای انجام این کار پیشنهاد کردند که در آن همه دنباله‌های کلمات مجاور استخراج می‌شوند. اگر دو سند دربرگیرنده مجموعه‌های شینگل یکسان باشند، آن‌ها را مشابه تلقی می‌کنند و اگر شینگل‌ها با هم تداخل یا تلاقی کنند، آن‌ها را مشابه‌های همسان می‌نامند. با این حال محققین دریافتند که

این روش برای اسناد کوچک عملکرد مناسبی ندارد. فترلی و همکارانش از ۵-گرم بعنوان یک شینگل^{۲۵}

استفاده کردند و نمونه‌ای از ۸۴ شینگل را برای هر سند مورد استفاده قرار دادند. سپس این ۸۴ شینگل در

شش فرا-شینگل قرار می‌گیرند. اسنادی که دارای دو فرا-شینگل مشترک باشند، اسناد شبه کپی برداری

شده تلقی می‌شوند. برودر روش مؤثری را برای تعیین شباهت ساختاری فایل‌ها ارائه کرده است و آن را برای

هر سند در شبکه جهانی وب (www) مورد استفاده قرار داده است. او با استفاده از این مکانیسم، دسته بندی

همه اسناد مشابه از نظر ساختاری را انجام داده است.

روش دیگری برای تشخیص و حذف صفحات وب شبه کپی برداری شده، که با عنوان اطلاعات متنی مبتنی

بر اولویت بندی شناخته می‌شود، در مقاله لینگ، هگزین و گواردیان پیشنهاد شده است. با این روش،

الگوریتمی برای استخراج اطلاعات متنی از صفحات وب توسط درخت DOM و یک الگوریتم اولویت محور

برای تشخیص اطلاعات متنی شبه کپی برداری شده اجرا می‌شوند تا نویز صفحات وب را کاهش دهند و در

نتیجه کارآمدی و بازدهی تشخیص اطلاعات متنی شبه کپی برداری شده را ارتقا بخشند.

نارایانا و همکارانش، روشی را برای تشخیص صفحات وب شبه کپی برداری شده در مرورگری وب پیشنهاد

کردند. صفحات وب شبه کپی برداری شده با ذخیره‌سازی صفحات مرورگری شده در منابعی شناسایی

می‌شوند. کلمات کلیدی از صفحات مرورگری شده استخراج شوند و بر مبنای این کلمات کلیدی، نمره

شباهت میان دو صفحه وب محاسبه می‌شود. در صورتی که نمره شباهت به یک حد آستانه برسد، اسناد

مورد نظر مشابه تلقی می‌شوند. با این حال، این محققین از اطلاعات ساختاری صفحات وب که برای دریافت اطلاعات معنایی پیرامون یک صفحه اهمیت دارند، استفاده نکردند.

متیو، شین داس و ویجایارانگاوان یک ایده جدید را برای تشخیص صفحات وب شبه کپی برداری شده پیشنهاد کردند که باز هم از الگوریتم مرحله درختی استفاده می‌کند و ارزیابی شباهت در این الگوریتم بر مبنای تجزیه مقدار منفرد (SVM) با استفاده از یک زاویه آستانه θ می‌باشد. با این حال SVM نیازمند عملکردهای ریاضی پیچیده‌تری در ماتریس TDW به همراه تبدیل آستانه Jaccard، t ، به زاویه آستانه θ می‌باشد. این مسئله باعث افزایش پیچیدگی‌های الگوریتم و مشکلات عملی در اندازه‌گیری زاویه می‌شود. متیو، شین داس و ویجایارانگاوان روش جدیدی با نام MWO را برای ارزیابی شباهت مورد استفاده قرار دادند که به طور مستقیم روی آستانه Jaccard، t ، کار می‌کند و پیچیدگی الگوریتم را کاهش می‌دهد. آن‌ها در این روش، ویژگی‌های معنایی، محتویات و ساختار یک صفحه وب را هم‌زمان بررسی می‌کنند. الگوی وزن گذاری پیشنهادی توسط خود آن‌ها در مقاله‌ای دیگر برای ایجاد یک ماتریس وزن گذاری سند TDW به کار گرفته می‌شود که نقش مهمی را در الگوریتم پیشنهاد دارد. یک الگوریتم سه مرحله‌ای مورد استفاده قرار گرفته است که یک رکورد ورودی را به همراه یک مقدار آستانه دریافت کند و یک مجموعه بهینه از شبه کپی برداری‌ها را به دست می‌دهد. در مرحله اول، که مرحله نمایش است، همه پیش پردازش‌ها انجام می‌شوند و یک الگوی وزنی مورد استفاده قرار می‌گیرد. سپس مرتب سازی به طور کلی به همراه روش‌های نرمال سازی استاندارد برای ایجاد یک ماتریس TDW انجام می‌شود. در مرحله دوم، که مرحله

فیلترینگ است، دو مکانیسم معروف فیلترینگ با عناوین فیلترینگ پیشوند و فیلترینگ محل قرارگیری برای کاهش اندازه مجموعه رقابتی رکورد به کار گرفته می‌شوند و در نتیجه تعداد مقایسه ها کاهش پیدا می‌کند. در مرحله سوم، که مرحله ارزیابی است، بررسی شباهت‌ها با ارائه روش جدیدی به نام تداخل وزنی مینیمم (MWO) بر مبنای یک مقدار آستانه به انجام می‌رسد، و در نهایت یک عدد بهینه از رکوردهای شبه کپی برداری شده حاصل می‌شود.

فصل پنجم : سرقت ادبی در کدهای کامپیوتری

سرقت ادبی در علوم کامپیوتر

سرقت ادبی در فایل‌های کد منبع زمانی اتفاق می‌افتد که کد منبع بدون اشاره به مالک و نویسنده اصلی آن کپی برداری و ویرایش شود. براساس گفته جوی و لاک، تغییرات لغوی در حقیقت تغییراتی هستند که می‌توانند در کد منبع بدون اثرگذاری بر تجزیه (parsing) برنامه انجام شوند و بر تجزیه کد اثر می‌گذارند و شامل باگ زدایی از برنامه می‌باشد. نمونه های تغییرات لغوی شامل تغییر نام شناساگرها و کامنت ها هستند، و از نمونه‌های تغییرات ساختاری نیز می‌توان به بازآرایی و جایگزینی جمله‌ها اشاره کرد.

سرقت ادبی کد منبع را میتوان بصورت انتقال بخشی از کد منبع نوشته شده توسط فردی دیگر به کد

شخصی (بدون اشاره به اینکه کدام بخش‌ها از محقق دیگری کپی شده اند) تعریف کرد.

سرقت ادبی معمولاً در محیط‌های آکادمیک اتفاق می‌افتد. دانشجویان به صورت عمدی یا غیرعمدی بخشی

از منبع را بدون ذکر نام، در اثر خود می‌گنجانند. تشخیص سرقت ادبی به صورت دستی در مجموعه‌ای از

صدها اثر غیرممکن و غیراقتصادی است. بدین ترتیب، ابزارهای نرم‌افزاری برای کمک به مدرسین در

تشخیص سرقت ادبی به‌وجود آمده‌اند.

باید دانست که هیچ یک از این ابزارها نمی‌توانند حقیقتاً وجود یا عدم وجود سرقت ادبی را نشان دهند. این

ابزارها صرفاً یک شاخص مشابهت را برای هر زوج از برنامه‌ها به دست می‌دهند و سپس بررسی‌های انسانی

برای تعیین این مسئله لازم است که این شباهت‌ها در حقیقت سرقت ادبی هستند یا آنکه صرفاً به واسطه

انجام یک تکلیف به صورت استاندارد و عرف مشاهده شده‌اند.

مطالب سرقت شده در کدها [۱۳]

سرقت ادبی در تکالیف برنامه‌نویسی می‌تواند فراتر از کپی کردن کد منبع^{۲۶} باشد؛ این کار ممکن است

دربرگیرنده کامنت‌ها، داده‌های ورودی برنامه، و طرح اینترفیس باشد.

فرآیند بررسی اینکه آیا سرقت ادبی در کد منبع انجام شده است یا خیر، می‌تواند شامل بررسی بخش‌های

دیگری از یک تکلیف برنامه‌نویسی باشد زیرا در برخی از موارد، کد منبع به‌تنهایی برای شناسایی و اثبات

سرقت ادبی کافی نیست. یک تکلیف برنامه‌نویسی می‌تواند شامل نمودارهای طراحی، کد منبع و سایر اسناد

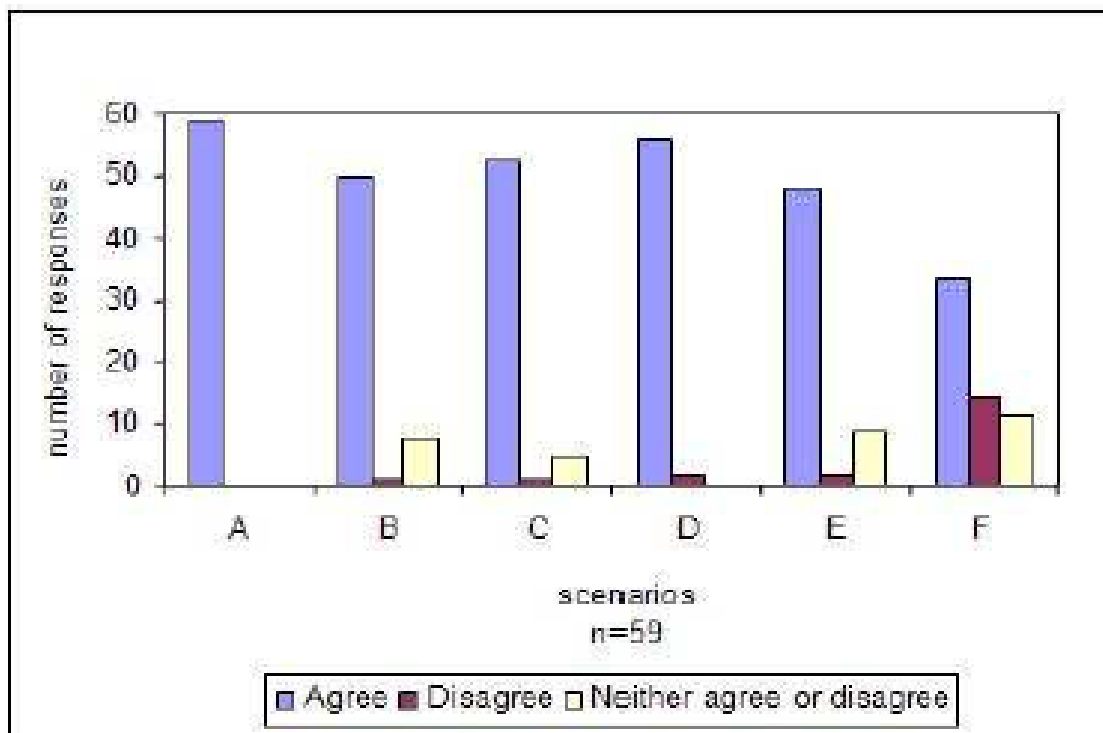
^{۲۶} Source Code

باشد. از افراد آکادمیک پرسیده شد که در هر یک از سناریو های زیر، یکی از پاسخ های «موافق»، «مخالف»، و «نه موافق نه مخالف»، را انتخاب کنند.

• سرقت ادبی در تکالیف برنامه نویسی می تواند شامل موارد زیر باشد:

- سناریو A: کد منبع یک برنامه کامپیوتری
- سناریو B: کامنت های^{۲۷} موجود در کد منبع
- سناریو C: مطالب طراحی یک برنامه کامپیوتری
- سناریو D: سند یک برنامه کامپیوتری
- سناریو E: اینترفیس کاربری یک برنامه کامپیوتری
- سناریو F: داده های ورودی برنامه، برای آزمایش برنامه

پاسخ های ارائه شده در تصویر ۶ آمده اند.



تصویر ۶: سناریو ها و پاسخ ها

همه افراد موافق بودند که در یک تکلیف برنامه نویسی، کد منبع می تواند مورد سرقت ادبی قرار گیرد. کامنت های موجود در کد منبع نیز می تواند سرقت شود و البته می تواند به شناسایی موارد سرقت کد منبع کمک کند. همچنین اغلب افراد توافق داشتند که کامنت ها نیز قابل سرقت هستند و همچنین ممکن است بتوانند به شناسایی موارد سرقت کد منبع کمک کنند.

داده های ورودی برنامه و اینترفیس کاربری نیز در صورتی که بخشی از الزامات تکلیف مورد نظر باشند، ممکن است مورد سرقت قرار گیرند. اغلب پاسخ دهندگان موافق بودند که داده های ورودی برنامه می توانند سرقت شوند، اما این مسئله به تنهایی به شناسایی سرقت ادبی کمکی نمی کند. سه نفر از افراد آکادمیک اعتقاد داشتند که کپی کردن داده های ورودی در هنگامی مسئله ساز می شود که دانشجویان از نظر

استراتژی‌های آزمایشی مورد سنجش قرار گیرند. هنگامی که این کار انجام می‌شود، ارزیابی سرقت ادبی با مشاهده استراتژی آزمایشی، شامل پایگاه‌های داده‌ای (مثلاً داده‌های ورودی) مورد استفاده برای آزمودن برنامه، و مطالب آزمایشی، شامل طرح آزمایش، سند طراحی سیستم، سند فنی و راهنماهای کاربری میسر خواهد بود.

طرح‌های مذنون مربوط به اینترفیس که توسط دانشجویان ارائه می‌گردند نیز باید از نظر سرقت مورد بررسی قرار گیرند. افراد عنوان کردند که این که یک اینترفیس و کاربری در معرض سرقت قرار گیرد یا خیر مبتنی بر این مسئله است که آیا در تکلیف مورد نظر، طراحی اینترفیس از دانشجو خواسته شده است یا خیر.

انواع سیستم‌های شناسایی سرقت ادبی

سیستم‌های شناسایی سرقت ادبی را می‌توان به انواع هرمتیک و وب، و از جنبه‌ای دیگر به انواع چندمنظوره، زبان طبیعی و کد منبع تقسیم‌بندی کرد. سیستم‌های شناسایی مربوط به وب تلاش میکنند که موارد انطباق مربوط به اسناد مذنون در منابع آن‌لاین را بیابند. سیستم‌های هرمتیک تنها در یک مجموعه محلی از اسناد، به دنبال نمونه‌های سرقت ادبی می‌گردند. این سیستم‌ها دارای یک پایگاه داده‌ای از اسناد هستند. برای مثال، پایگاه داده‌ای ممکن است دربرگیرنده آثار سایر محققین و نیز مطالب مورد استفاده در یک واحد درسی خاص باشد.

ابزارهای شناسایی سرقت ادبی کد منبع

ابزارهای مختلفی برای شناسایی سرقت ادبی وجود دارند که می‌توانند بر مبنای الگوریتم، تقسیم بندی شوند. در این بخش به توصیف ابزارهای شناسایی سرقت ادبی کد منبع و با استفاده از دسته‌بندی‌های

معرفی شده توسط موزگووی می‌پردازیم. آن‌ها شامل سیستم‌های مبتنی بر فینگرپرینت، و روش‌های مقایسه محتویات هستند. دسته‌بندی‌های مختلف دیگری نیز در مقالات وجود دارند.

سیستم‌های مبتنی بر فینگرپرینت (اثر انگشت)

ابزارهایی که بر اساس فینگرپرینت کار می‌کنند، به ایجاد فینگرپرینت‌هایی برای هر فایل می‌پردازند، که شامل اطلاعات آماری مربوط به فایل می‌باشد-مانند تعداد عبارات در هر خط، تعداد عبارات منحصر به فرد، تعداد کلمات کلیدی. اگر فینگرپرینت پروفایل نزدیک به هم باشد، آن‌ها یکسان تلقی می‌شوند. این نزدیکی معمولاً به واسطه سنجش فاصله میان آن‌ها (با استفاده از یک تابع فاصله از نظر ریاضی) تعیین می‌شود. در سال‌های ابتدایی، فینگرپرینت‌ها در سیستم‌های «شمارش مشخصه» به کار گرفته می‌شدند. اولین سیستم شناخته شده تشخیص سرقت ادبی یک برنامه شمارش مشخصه بود که توسط اوتن اشتاین برای شناسایی آثار یکسان یا نزدیک به هم مربوط به دانشجویان پدید آمد. این برنامه از شاخص‌های نرم‌افزاری هالستد برای شناسایی سرقت ادبی به واسطه شمارش عملگرها و عملوندها (operator and operand) برای مدول‌های ANTI-FORTRAN استفاده کرد. شاخص‌های به کار گرفته شده توسط هالستد عبارت بودند از:

- تعداد عملگرهای^{۲۸} منحصر به فرد
- تعداد عملوندهای منحصر به فرد
- تعداد کلی وقوع عملگرها
- تعداد کلی وقوع عملوندها^{۲۹}

^{۲۸} Operator

رابینسون و سوفایک برنامه تشخیص سرقت ادبی را به کار گرفتند که ترکیبی بود از شاخص‌های جدید و شاخص‌های هالستد به منظور بهینه‌سازی شناسایی سرقت ادبی. سیستم آن‌ها با نام ITPAD متشکل از سه مرحله بود: آنالیز لغوی، آنالیز ساختار برنامه برای ویژگی‌ها، و آنالیز ویژگی‌ها. ITPAD (ابزار سازمانی برای اصلاح برنامه) هر برنامه را به چند بلوک تقسیم می‌کند و یک گراف را برای نمایش ساختار برنامه می‌سازد. سپس فهرستی از ویژگی‌ها را بر مبنای آن لیست‌های لغوی و ساختاری پدید می‌آورد و زوج‌های برنامه‌ها را با شمارش این ویژگی‌ها مقایسه می‌کند.

رامبالی و سیج یک سیستم شمارش مشخصه را پدید آوردند که برنامه‌های دانشجویان را می‌گیرد، آن‌ها را تفکیک (parse) می‌کند و سپس یک «سیستم اطلاعات» را می‌سازد که دربرگیرنده بردارهای دانش است و در آن هر بردار اطلاعاتی در مورد مشخصه‌ها در برنامه یک دانشجو را در اختیار می‌گیرد. این محققین از ویژگی‌های مشابه شناسایی شده توسط سیستم‌های پیش‌گفته استفاده می‌کنند، با این حال از روش مجزایی بهره می‌برند که ویژگی‌های حلقه-ساز^{۲۰} را در یک برنامه می‌شمارد. به جای شمارش گونه‌های جملات حلقه-ساز به صورت مجزا، آن‌ها در میان انواع حلقه‌ها تفکیک قائل نمی‌شوند، و شمارش همه این جملات را در شمارش یک مشخصه در نظر می‌گیرند. هنگامی که بنیان‌های دانش ایجاد می‌شوند، برنامه‌ها در یک «درخت تصمیم» دسته‌بندی می‌گردند. بر مبنای درخت تصمیم، برنامه‌هایی که حاوی شباهت‌هایی هستند، شناسایی خواهند شد.

از آن زمان، ابزارهای پیشرفته‌ترین برای شناسایی سرقت ادبی به وجود آمدند. این ابزارها در مقالات عموماً با

عنوان «سیستم‌های شاخص ساختاری» شناخته می‌شوند. سیستم‌های ساختاری مقایسه‌ای را در مورد

ساختار برنامه‌ها از نظر شباهت انجام دهند. موزگووی آن‌ها را عنوان «روش‌های مقایسه محتویات» دسته

بندی کرده است.

مقایسه سیستم‌های مبتنی بر انطباق رشته و شمارش مشخصه نشان داده است که روش‌های شمارش

مشخصه به‌تنهایی برای شناسایی سرقت ادبی کافی نیستند. ابزارهای جدید در شناسایی سرقت ادبی مانند

MOSS، ترکیبی از روش‌های فینگرپرینت با روش شاخص ساختاری است.

MOSS (معیار شباهت نرم‌افزاری) در سال ۱۹۹۴ توسط آیکن و برکلی بعنوان سیستمی برای بررسی شباهت

های کد منبع نوشته شده در C، C++ و پاسکال بوجود آمد [۱۴]. مبتنی است و یک الگوریتم انطباق

رشته که به واسطه تقسیم برنامه به k-گرم‌ها عمل می‌کند، که در آن k-گرم یک زیر-رشته پیوسته با طول

k است. هر k-گرم درهم (hash) می‌شود و MOSS زیرمجموعه از این مقادیر درهم را به صورت فینگرپرینت

برنامه انتخاب می‌کند. شباهت به واسطه تعداد فینگرپرینت‌های مشترک در برنامه‌ها تعیین می‌شود؛ یعنی

هرچه فینگرپرینت‌های مشترک بیشتری وجود داشته باشد، آن‌ها مشابه تر هستند. برای هر زوج از قطعه

های کد منبع شناسایی شده، خلاصه نتایج شامل تعداد نشانه‌های (token) منطبق، و در صد تداخل کد منبع

بین زوج‌های فایل‌ها است.

روش‌های مقایسه محتویات

روش‌های مقایسه محتویات معمولاً با عنوان سیستم‌های شاخص ساختاری در مقالات شناخته می‌شوند. این

سیستم‌ها، برنامه‌ها را به نشانه‌ها تبدیل می‌کنند و سپس دنباله ای پیوسته و منطبق از زیر-رشته‌های

موجود در برنامه‌ها را جستجو می‌کنند. شباهت میان برنامه‌ها مبتنی است بر در صد متن‌های منطبق.

موزگووی روش‌های مقایسه محتویات را به الگوریتم‌های مبتنی بر انطباق رشته، الگوریتم‌های انطباق

پارامتری سازی شده، و الگوریتم‌های مقایسه درخت‌های تجزیه (parse trees) تقسیم می‌کند.

الگوریتم‌های انطباق رشته^{۳۱}

جدید ترین سیستم‌های تشخیص سرقت ادبی متکی هستند بر مقایسه ساختار برنامه‌ها. این سیستم‌ها از

شاخص‌های شمارش مشخصه استفاده می‌کنند اما همچنین ساختار برنامه را نیز به منظور بهبود فرآیند

تشخیص سرقت ادبی، مقایسه می‌نمایند. سیستم‌های مبتنی بر انطباق رشته از الگوریتم‌های مقایسه‌ای

استفاده می‌کنند که پیچیده‌تر از الگوریتم‌های شمارش مشخصه هستند. اغلب الگوریتم‌های انطباق رشته

واسطه تبدیل الگوریتم‌ها به نشانه‌ها عمل می‌کنند و سپس از یک الگوریتم پیچیده جستجو برای شناسایی

زیر-رشته‌های مشترک متنی بین و برنامه استفاده می‌نمایند.

این سیستم‌ها ابتدا توسط دونالدسون پیشنهاد شدند. این محقق روش‌های ساده‌ای را شناسایی کرد که

دانشجویان تازه کار برنامه‌نویسی برای تشخیص سرقت ادبی از آن‌ها استفاده می‌کنند. این روش‌ها عبارتند

از:

^{۳۱} String Matching

- تغییر نام متغیرها

- بازآرایی جملاتی که بر نتیجه برنامه اثر گذار نیستند

- تغییر فرمت جملات

- شکستن جملات، مثلاً تعریف‌های چندگانه و جملات خروجی

روش‌های شناسایی شده توسط دونالدسون اساسی‌ترین اشکال حمله هستند. ویل و جوی و لاک نیز فهرست

دقیقی از حملات را ارائه کرده‌اند. یک فهرست بسیار کامل‌تر توسط پرچلت مطرح شد. ما همچنین یک

فهرست جامع را که دربرگیرنده حملات سرقت ادبی هستند ارائه می‌کنیم.

برنامه ارائه شده توسط دونالدسون به اسکن فایل‌های کد منبع می‌پردازد و اطلاعات مربوط به انواع خاصی از

حملات را ذخیره سازی می‌کند. سپس یک کاراکتر منفرد کد به انواع جملاتی که در توصیف ساختار اهمیت

دارند، اختصاص می‌یابند. سپس هر تخصیص به صورت رشته‌ای از کاراکترها نمایش داده می‌شود. اگر

نمایش رشته‌ها یکسان یا مشابه باشد، زوج برنامه‌های مورد بررسی یکسان تلقی می‌شوند.

برخی از سیستم‌های جدید مبتنی بر انطباق رشته شامل Plague^۳، Yap^۳ (که خود نوع دیگری از Plague

است)، JPlag و Sherlock هستند [۱۵].

در اغلب سیستم‌های مبتنی بر انطباق رشته، جمله موارد پیش گفته، مرحله اول نشانه سازی (tokenization)

نامیده می‌شود. در مرحله نشان سازی، هر فایل کد منبع از نشانه‌های از پیش تعیین شده و منسجم

جایگزین می‌شود؛ مثلاً انواع مختلف حلقه‌ها در کد منبع می‌تواند با نام نشانه مشابهی از همان نوع حلقه

جایگزین شود (مثلاً حلقه while، حلقه for). پس هر فایل کد منبع نشانه سازی شده با یک سری رشته‌های

نشانه‌ای نمایش داده می‌شود. سپس برنامه‌ها از طریق جستجو برای دنباله‌های زیر-رشته‌ای انطباقی

مربوط به نشانه‌ها مقایسه می‌شوند. برنامه‌هایی که در برگیرنده تعدادی نشانه‌های انطباقی در بالای یک

آستانه معین باشند، یکسان تلقی می‌شوند. شباهت میان دو فایل اساساً به واسطه پوشش نشانه‌های انطباقی

میان فایل‌های شناسایی شده محاسبه می‌شود.

Plague ابتدا ترتیبی از نشانه‌ها را برای هر فایل ایجاد می‌کند و سپس نسخه‌های نشانه‌ای برنامه‌های انتخابی

را با استفاده از یک روش انطباق رشته، مقایسه می‌نماید. نتایج به صورت فهرستی از زوج‌های انطباقی

نمایش داده می‌شوند که با ترتیب درجه شباهت مربوط به طول بخش انطباقی از دنباله‌های نشانه میان دو

فایل ارائه شده‌اند. با این حال مشکلاتی در زمینه Plague مطرح گشته‌اند. یکی از مشکلات Plague آن است

که تبدیل آن به یک زبان برنامه‌نویسی دیگر وقت‌گیر است. بعلاوه، نتایج در دو فهرستی نشان داده می‌شوند

که به واسطه شاخص‌هایی مرتب شده‌اند که تفسیر نتایج را مشهود نمی‌سازند. در آخر اینکه Plague چندان

بازدهی بالایی ندارد زیرا مبتنی بر تعدادی از ابزارهای الحاقی Unix که باعث ایجاد مشکلاتی در قابلیت

انتقال می‌شود.

YAP^۳ برنامه‌ها را به رشته‌ای از نشانه‌ها تبدیل می‌کند و آن‌ها را با استفاده از یک الگوریتم انطباق نشانه،

الگوریتم RKR-GST به منظور یافتن قطعات مشابه کد منبع، مقایسه می‌نماید. YAP^۳ فایل‌های کد منبع را

پیش از تبدیل به نشانه‌ها، پیش پردازش می‌کند. این پیش پردازش شامل حذف کامنت‌ها، تبدیل حروف

بزرگ به حروف کوچک، نگاشت مترادف ها به یک شکل مشترک (یعنی تابع به روش نگاشت می‌شود)، بازآرایی توابع در ترتیب فراخوانی آن‌ها، و حذف همه نشانه‌هایی که مربوط به لغات زبان هدف نیستند (یعنی حذف همه عباراتی که جز عبارات محفوظ برای زبان نیستند). YAP^۳ اساساً برای تشخیص شکستن توابع کد به توابع متعدد، و نیز برای شناسایی بازآرایی قطعه‌های مستقل کد منبع بوجود آمد. این الگوریتم به واسطه مقایسه دو رشته (الگو و متن) کار می‌کند که دربرگیرنده جستجوی متنی برای یافتن زیر-رشته‌های انطباقی الگو می‌باشد. بخش‌های انطباقی زیر-رشته‌ها، موزاییک (tile) نامیده می‌شوند. هر موزاییک در حقیقت یک انطباق است که دربرگیرنده یک زیر-رشته از الگو و یک زیر-رشته از متن می‌باشد. هنگامی که انطباقی یافت شد، موقعیت نشانه‌های موزاییک‌ها تنظیم می‌شود. از موزاییک‌هایی که طول آن‌ها در زیر یک آستانه طول انطباق مینیمم است صرف‌نظر می‌شود. هدف از الگوریتم RKR-GST یافتن انطباقی‌های ماکسیمال دنباله‌های زیر-رشته پیوسته است که دربرگیرنده نشانه‌هایی است که توسط زیر-رشته‌های دیگر پوشش داده نشده است و در نتیجه تعداد نشانه‌های پوشش داده شده توسط موزاییک‌ها به حداکثر می‌رسد.

JPlag نیز از الگوریتم مقایسه‌ای مشابه YAP^۳ استفاده کند اما با بازدهی زمان اجرای بهینه. در JPlag، مشابهت به واسطه در صد رشته‌های نشانه‌ای پوشش داده شده محاسبه می‌شود. یکی از مشکلات JPlag آنست که فایل‌ها باید تجزیه شوند تا در مقایسه مربوط به سرقت ادبی گنجانده شوند، و این مسئله باعث از دست رفتن فایل‌های مشابه می‌شود. همچنین پارامتر طول انطباق مینیمم که در JPlag توسط کاربر تعریف می‌شود، در یک عدد پیش‌فرض قرار داده می‌شود. تغییر این عدد می‌تواند نتایج شناسایی را بهتر یا بدتر کند، و برای تغییر این عدد، نیاز به درکی از الگوریتم ماورای JPlag وجود دارد (یعنی RKR-GST). JPlag به

صورت یک سرویس وب اجرا می‌شود و در برگیرنده یک اینترفیس کاربری ساده اما مؤثر است. اینترفیس

کاربری نشان‌دهنده فهرستی از زوج‌های فایلی مشابه و درجه شباهت آن‌ها است، و همچنین شامل نمایش

مقایسه‌ای از فایل‌های مشابه شناسایی شده بواسطه برجسته سازی بلوک‌های انطباقی آن‌ها از قطعات کد

منبع می‌باشد.

Sherlock نیز یک الگوریتم مشابه YAP^۳ را اجرا می‌کند. Sherlock برنامه‌ها را به نشانه‌ها تبدیل می‌کند و

دنبال‌های خطوطی را (که run) نامیده می‌شوند جستجو می‌کند که در دو فایل مشترک هستند. همانند

الگوریتم YAP^۳ ، Sherlock نیز به دنبال run‌های با طول مشابه می‌گردد. اینترفیس کاربری Sherlock فهرستی

از زوج‌های فایل‌های مشابه و درجه شباهت آن‌ها را نمایش می‌دهد و بلوک‌های انطباقی قطعه‌های کد منبع

آن‌ها را که در زوج فایل‌های شناسایی شده یافت شده‌اند، مشخص می‌کند. به‌علاوه، Sherlock نمایش سریعی

از نتایج را به شکل یک گراف ایجاد می‌کند که در آن هر رأس بیان‌کننده یک فایل منفرد کد منبع و هر

گوشه نشان‌دهنده درجه شباهت میان دو فایل است. گراف تنها شباهت (گوشه‌های) میان فایل‌ها در بالای

یک سطح تعریف شده توسط کاربر می‌باشد. یکی از مزایای Sherlock آن است که برخلاف JPlag، نیازی به

تجزیه فایل‌ها به منظور گنجاندن آن‌ها در مقایسه وجود ندارد و هیچ پارامتر تعریف شده توسط کاربر، که بر

عملکرد سیستم اثرگذار باشد، موجود نیست. Sherlock یک ابزار منبع باز است و روش انطباقی نشانه آن

به‌راحتی قابل هماهنگ سازی با زبان‌های دیگری به جز Java می‌باشد. Sherlock ابزار مستقل است و یک

سرویس مبتنی بر وب مانند JPlag و MOSS نیست. یک ابزار مستقل می‌تواند برای افراد آکادمیک از نظر

بررسی فایل‌های دانشجویی به دنبال سرقت ادبی، با مدنظر قرار دادن مسائل محرمانگی، مطلوب‌تر باشد.

Plaggie ابزاری است همانند JPlag اما بدون الگوریتم‌های بهینه‌سازی سرعت. Plaggie فایل‌ها را به نشانه‌ها

تبدیل می‌کند و از الگوریتم RKR-GST برای مقایسه فایل‌ها از نظر شباهت استفاده می‌کند. ایده ماورای

Plaggie ایجاد ابزار همانند JPlag است که بتواند مستقل باشد (یعنی بتواند روی یک دستگاه محلی نصب

شود) و یک ابزار مبتنی بر وب نباشد، و اینکه دارای کارکردهای بیشتری بوده و افراد آکادمیک را قادر

سازد که کد را از طریق مقایسه استخراج کنند.

یک ابزار دیگر شناسایی سرقت ادبی که توسط موزگووی مطرح شد، سیستم تشخیص سریع سرقت ادبی

(FDPS) بود که هدف آن بهسازی سرعت تشخیص سرقت ادبی با استفاده از یک ساختار داده شاخص

گذاری شده برای ذخیره‌سازی فایل‌ها بود. در ابتدا فایل‌ها به نشانه‌ها تبدیل می‌شوند و فایل‌های نشانه

گذاری شده در یک ساختار داده‌ای شاخص بندی شده قرار می‌گیرند. این مسئله امکان جستجوی سریع

فایل‌ها را با استفاده از الگوریتمی مشابه الگوریتم مورد استفاده در YAP^۳ فراهم می‌آورد. این کار شامل اتخاذ

یک «فایل آزمایشی» و جستجو بدنبال زیر-رشته انطباقی در «فایل مجموعه» می‌باشد. موارد انطباقی در یک

مخزن ذخیره سازی می‌شوند و سپس برای محاسبه شباهت‌های میان فایل‌ها مورد استفاده قرار می‌گیرند.

شباهت عبارتست از نسبت تعداد کلیه نشانه‌های منطبق در فایل مجموعه به تعداد کلیه نشانه‌ها در فایل

آزمایشی. این نسبت می‌تواند پوشش (جامعیت) نشانه‌های منطبق با تعیین کند. زوج‌های فایل‌هایی دارای

یک مقدار شباهت بالای یک حد آستانه معین بازیابی می‌شوند. یکی از مشکلات ابزار FDPS آن است که

نتایج آن را نمی‌توان به واسطه نمایش قطعه‌های کدهای مشابه مشاهده کرد. بدین ترتیب، محققین FDPS

را با ابزار Plaggie ترکیب کردند تا مقایسه فایل به فایل را انجام داده و نتایج را مشاهده کنند.

الگوریتم های انطباقی پارامتری سازی شده

الگوریتم های انطباق پارامتری سازی^{۳۲} شده همانند روش های متداول انطباق نشانه هستند، اما نشانه ساز

پیشرفته تری دارند. اساساً این الگوریتم های انطباق پارامتری سازی شده با تبدیل فایل ها به نشانه ها کار

می کنند. یک انطباق پارامتری (که آن را انطباق - p نیز می نامند)، قطعه های کد منبعی را که نام های متغیر

آن ها به صورت سیستماتیک جایگزین شده است (تغییر نام) را منطبق می نمایند.

ابزار **Dup** مبتنی است بر یک الگوریتم انطباق - p و برای شناسایی کد کپی شده در نرم افزار ارائه شده است.

این ابزار می تواند بخش های یکسان و پارامتری سازی شده کد منبع را شناسایی کند. در ابتدا، با استفاده از

یک آنالیزور لغوی، ابزار به اسکن فایل کد منبع می پردازد و برای هر خط از کد، یک خط تبدیل شده ایجاد

می شود. کد منبع به پارامترها تبدیل می شود؛ و این فرایند شامل تبدیل شناساگر ها و ثابت ها به سمبل

(علامت) مشابه p، و فهرستی از کاندیداهای پارامتر است. برای مثال خط $x = fun(y) + r * x$ به $P = P(P) + P * P$

تبدیل می شود و فهرستی شامل x ، y ، r ، fun و x ایجاد می شود. سپس آنالیزور لغوی یک عدد صحیح را

ایجاد می کند که هر خط کد تبدیل شده را نشان می دهد. با داشتن یک طول آستانه، Dup به یافتن انطباق

های p کد منبع می پردازد. شباهت به واسطه محاسبه انطباق های p میان دو فایل تعیین می شود. Dup در

صد شباهت میان دو فایل را به صورت خروجی ارائه می کند، یک پروفایل که نشان دهنده خطوط کد

انطباق یافته p است و یک نمودار که نشان دهنده موقعیت کد انطباقی می باشد. با این حال، بنا به گفته

گیتچل و تران، این را در صورت درون گذاری جملات کاذب و یا در صورت بازآرایی بلوک کوچکی از

جملات ، نمی تواند فایل ها را شناسایی کند. همچنین الگوریتم های انطباق پارامتری دیگری نیز در مقالات وجود دارند.

الگوریتم های مقایسه درخت های تجزیه [۱۶]

الگوریتم های مقایسه درخت تجزیه^{۳۳} می توانند روش های مقایسه ای را به کار گیرند که ساختار فایل ها را با هم مقایسه می کنند. برنامه کاربردی کمی SIM هر فایل را به صورت یک درخت تجزیه نمایش می دهد و سپس با استفاده از نمایش رشته ای درخت های تجزیه ، زوج های فایل ها را از نظر مشابهت با هم مقایسه می کند. فرایند مقایسه با تبدیل هر فایل کد منبع به یک رشته از نشانه ها آغاز می شود که در آن هر نشانه با یک مجموعه ثابت ثابت از نشانه ها جایگزین می گردد. پس از نشانه گذاری (نشانه سازی)^{۳۴} ، جریان های نشانه ای فایل ها به بخش های مختلف تقسیم می شوند و این بخش ها هم تراز می گردند. این روش امکان شناسایی قطعه های کد به هم ریخته را فراهم می آورد.

SIM از الگوریتم های انطباق رشته معمولی برای مقایسه زوج های فایل های نشان داده شده به صورت درخت های تجزیه استفاده می کند. همانند اغلب الگوریتم های انطباق رشته ای، SIM نیز زیر- دنباله نشانه های مشترک ماکسیمال را جستجو می کند. درجه شباهت میان زوج های فایل ها (در دامنه ۰,۰ تا ۱,۰ (به صورت خروجی ارائه می شود. SIM برای شناسایی شباهت در برنامه های C ایجاد شده است. براساس گفته سازندگان SIM، این برنامه کمکی به راحتی قابل انطباق با برنامه های نوشته شده به زبان هایی به جز

^{۳۳} Parse Tree
^{۳۴} Tokenization

C است ؛ و می تواند فایل های مشابه دربرگیرنده اصلاحات و تغییرات رایجی مانند تغییر نام، بازآرایی توابع و جملات، و افزودن/حذف کامنت ها و فضاهای خالی را شناسایی کند.

یکی از کاربردهای اخیر الگوریتم های مقایسه درخت های تجزیه چیزی است که در سیستم Brass اجرا می شود. کاربرد Brass مستلزم اجرای هر برنامه بصورت یک «جدول ساختاری» است ، که یک نمایش گرافیکی متشکل از نمایش درختی هر فایل است. ریشه درخت تعیین کننده هدر (header) فایل است، و گره های child تعیین کننده جملات و یک جدول علائم (دیکشنری داده ای) که دربرگیرنده اطلاعاتی در مورد متغیرها و ساختارهای داده مورد استفاده در فایل است، تبدیل می شود. مقایسه در Brass یک فرایند سه بخشی است. با الگوریتم اول دربرگیرنده مقایسه درخت های ساختاری فایل ها و الگوریتم سوم دربرگیرنده مقایسه دیکشنری داده ها مرتبط با فایل ها است. در ابتدا، زوج های فایل های مشابه با استفاده از الگوریتم مقایسه اول شناسایی می شوند. پس از آن، الگوریتم های مقایسه دوم و سوم برای زوج های فایل های شناسایی شده اعمال می شوند.

مقایسه درختی مستلزم استفاده از الگوریتم هایی پیچیده تر از الگوریتم های انطباق رشته ای است . با توجه به این مسئله می توان فرض کرد که سیستم های مبتنی بر سه الگوریتم مقایسه ای، کندتر از سیستم های مبتنی بر الگوریتم های انطباق رشته ای هستند. روش فیلترینگ Brass فرایند مقایسه را تسریع می کند.

براساس گفته موزگووی ، تحقیقات زیادی در حوزه الگوریتم‌های درخت‌های تجزیه انجام نشده است و

مشخص نیست آیا این الگوریتم با عملکردی بهتر از الگوریتم‌های انطباق رشته‌ای از نظر شناسایی زوج‌های

فایل‌های مشابه کد منبع دارند یا خیر.

Marble: از آنجاکه هیچ مقاله انگلیسی در مورد Marble منتشر نشده است در اینجا ویژگی‌های این ابزار را

بیشتر توصیف می‌کنیم.

Marble ابزاری است که در سال ۲۰۰۲ در دانشگاه اولترخت طراحی شده [۱۷] . هدف از آن ایجاد یک ابزار

ساده با قابلیت پشتیبانی آسان بود که می‌توانست برای شناسایی شباهت‌های مذنون بین کدهای Java مورد

استفاده قرار گیرد. با جمع‌آوری همه برنامه‌های ارائه شده برای کارکردهای مختلف در دپارتمان علوم

کامپیوتر در اولترخت و به واسطه مقایسه در میان آن‌ها، Marble در نشان دادن سرقت ادبی کارکرد مناسبی

را ارائه می‌کرد. به واسطه مقیاس پذیری، بسیار مهم است که این ابزار می‌تواند بین کدهای قدیمی و جدید

تمایز قائل شود.

Marble از یک روش ساختاری برای مقایسه کدها استفاده می‌کند. این کار با تفکیک کد به فایل‌هایی آغاز

می‌شود به گونه‌ای که هر فایل تنها دربرگیرنده یک دسته سطح بالا باشد. مرحله بعد، مرحله نرمالسازی

است تا جزئیاتی از این فایل‌ها که به راحتی توسط دانشجویان قابل تغییر است، حذف شوند: یک آنالیز لغوی

انجام میشود که کلمات کلیدی (مانند class، for) و دسته پرکاربرد و نام‌های روش (مانند String، System،

toString) را حفظ می‌کند. کامنت‌ها، فضاهای خالی اضافی، ثابت‌های رشته و تعریف‌های مهم حذف

می‌شوند، سایر نشانه‌ها به نشانه آن‌ها "تایپ"، انتزاع می‌گردند. برای مثال، هر عدد مبنای ۱۶ با H و هر کاراکتر لفظی با L جایگزین می‌شوند.

برای هر فایل، Marble دو نسخه نرمال سازی شده را محاسبه می‌کند: یکی که مرتبه فیلدها، روش‌ها و دسته‌های داخلی در آن دقیقاً مشابه فایل اصلی است، و یکی که در آن فیلدها، روش‌ها و دسته‌های داخلی با یکدیگر گروه بندی می‌شوند و این گروه‌ها مرتب می‌گردند. مرتب سازی به صورت شهودی انجام می‌شود. برای مثال، روش‌های تعداد از طریق تعداد نشانه‌ها، سپس به واسطه طول کلی رشته نشانه‌ها، و در نهایت به صورت الفبایی، مرتب می‌گردند .

به منظور استخراج دسته‌های داخلی، روش‌ها و فیلدها از فایل دسته Java بدون الزام به تجزیه، فایل Marble ابتدا به آکولادهای { و } در برنامه به عمق تودرتوی آن‌ها تفسیر می‌شود، و دسته‌ها به واسطه انطباق با آکولادهای عمق تودرتوی راست تفکیک می‌گردند (عمق * متناظر با آکولادهای تعریف دسته و عمق ۱ متناظر با روش‌ها و دسته‌های داخلی هستند). دانستن موقعیت آکولاد باز یک روش نمی‌تواند به طور مستقیم موقعیت آغازین روش به دست دهد، بلکه یافتن آن تنها با اسکن رو به عقب به سمت اولین نقطه-ویرگول یا آکولاد بسته امکان‌پذیر است. دسته‌های داخلی نیز به صورت مشابه پردازش می‌شوند.

در هنگام اجرای ابزار می‌توان مقایسه را در میان نسخه‌های نرمال سازی شده مرتب شده یا مرتب نشده (یا هر دو) انجام داد. از آنجا که مرتب سازی شهودی است، تغییرات کمی در روش در یک دسته می‌تواند به طور کامل مرتب سازی روش‌ها را تغییر دهد. موردی مشاهده شده است که در آن یک دانشجو تعدادی از

تغییرات را انجام داده بودم و روش‌ها را بازآرایی نکرده بود. به دلیل تغییرات، روش‌های متناظر در نسخه اصلی و نسخه سرقت شده دارای موقعیت‌های بسیار متفاوتی بودند که باعث تأثیرگذاری منفی بر اسکور شد. به همین دلیل، مقایسه نسخه‌های مرتب نشده نیز منطقی است.

فهرست روش‌های شناسایی کدهای کپی شده [۱۳]

در اینجا جدولی از روشهای مطرح برای شناسایی کدهای کپی شده را بصورت زیر ارائه می کنیم.

Tool	Citations	Sup. Lang.	Domain	Approach	B.Ground	L.Input	Validated?
Dup	Baker [14, 18]	c, c++, Java	CD/Unix	line-based/text-based	academic	1.1M LOC	With two systems
JPlag	Prechelt et al. [192]	Java, c, c++, Scheme, NL text	PD/Online	Token/Greedy String	Academic	236 LOC	Student assignments/artificial data
CloneDr	Baxter et al. [31, 30]	c, c++, Java, COBOL, Others with DMS domain	CD WindowNT	AST/Tree Matching	Commercial	400K LOC, C Code	Process Control System
DupLoc	Ducasse et al. [74, 72]	Language Independent/ 45 Mins to adapt	CD/Visual Works2.5	Line/Exact string matching	Academic	46K LOC	With 4 systems of different languages
CCFinder	Kamiya et al. [122]	C, C++, Java, COBOL and other with lexical analyzer and transformation rules	CD/Windows/NT	Transformation/Token comp. with suffix tree	Academic	10M LOC	With 4 systems
CP-Miner	Li et al. [168, 169]	C, C++	CD & Copy-pasted bugs detection /Windows/Linux	Sequence Database/Frequent subsequent mining	Academic	4365K LOC	Several systems
Sim	Gitchell et al. [90]	C	PD/Linux	Parse tree to string / String alignment	Academic	3.5K bytes	With 65 student assignments
Covet/CLAM	Mayrand [178]	C, Others supported by Datrix	CD	Metrics from Datrix, 4 Points of comp., Ordinal scale of 8 cloning level	Academic	507K LOC	With two telecommunication systems
DiLucca Pro.	Di Lucca et al. [67, 66]	HTML client & ASP server pages	Duplicated web-pages/PD	Sequence of tags/ Levenshtein distance	Academic	331 files	With 3 web applications
eMetrics	Fabio et al. [46, 161]	HTML & scripting languages	Visual inspection of potential function clones	Gets potential function clones from eMetrics	Academic	403 files	Validated with 4 applications

Konto's Tool	Kontogiannis et al. [146]	C, (other possible?)	System clustering, redocumentation & program understanding	Direct comparison of metrics values and Dynamic programming approach of begin-end block	Academic	>300K LOC	With several systems
codiml (Bauhaus)	Bellon [35, 35]	C, C++	CD/Linux	AST/Tree matching	Academic	235K LOC?	With several systems?
Google	Sager et al. [202]	Java	Finding similar java class	AST to FAMIX & Tree matching	Academic	?	Eclipse plug-in and test cases
Deckard	Jiang et al. [116]	C, Java and others with formally specified grammar, YACC grammar	CD/Linux	Tree, Characteristics vectors, Euclidean space	Academic	5,287K LOC	With Linux kernel and JDK
cpdetector	Koschke et al. [153, 154, 152]	C, C++	CD/Linux?	AST to sequence of AST tokens, suffix tree	Academic	235K LOC	With 4 systems and 8 tools
Duplix	Krinke et al. [156, 157]	C?	CD	PDG, graph matching	Academic	25K LOC	several test cases
clones	Koschke [153]	C, C++, Java, COBOL, VB	CD	Tokens/Suffix tree	Academic	235K LOC	With 4 systems and 8 tools
cscope	Frenzel [153]	C, C++	CD	Tokens/Suffix Tree/Post processing for syntactic clones	Academic	235K LOC	With 4 systems and 8 tools
PDG-DUP	Komondor & Horwitz [140, 141, 142]	C, C++?	CD / Clone Refactoring	PDG/ Slicing	Academic	?	With some systems
Tairas's Tool	Tairas et al. [206]	C, C++, C#, Phoenix framework languages?	CD/Exact function clones	AST/Suffix tree	Academic	1500K LOC	With several systems
RIF	Hasit et al. [24]	C	CD	Token/Suffix-array	Academic	3025K LOC	With Linux parts

جدول ۷: روش‌های رایج مورد استفاده برای شناسایی سرقت ادبی در کدهای کامپیوتری

مزایا و معایب	روش
روشهای فینگرپرینت نتایج مناسبی را بدست می دهند اما در صورتی که بخش سرقت شده تغییر کرده باشد (مثلاً کلمات جابجا شده یا جایگزین شده باشند) با مشکل مواجه میگردد.	فینگرپرینت
روش های مبتنی بر رشته ها قادر هستند مخدوش سازی های پیچیده بخش سرقت شده (مثل بازآرایی، ادغام، تفکیک جملات، و تغییر مفهومی کلمات) و نیز کپی برداری های مستقیم را تشخیص دهند. این روشها عملکرد سریعی دارند اما در عین حال ممکن است بواسطه تغییر نام شناساگرها ی اغتشاش و سردرگمی در آنها رخ دهد	الگوریتم های انطباق رشته ای
JPlag بعنوان یک سرویس وب در دسترس است و یک اینترفیس کاربری بسیار توانمند برای فهم نتایج دارد. این ابزار از نظر منابع پر بازده عمل میکند و برای سرقت های گسترده میتواند بشکلی مناسب مورد استفاده قرار گیرد. نقص اصلی JPlag آنست که مستلزم تجزیه مجموعه های داده ای است و اگر یک برنامه نتواند تجزیه شود، آن را از مجموعه داده ای حذف میکند.	JPlag
درخت های تجزیه میتوانند امکان مقایسه سطح بالا را فراهم کنند. مثلاً میتوانند جملات شرطی را نرمال سازی کنند و ساختارهای هم-ارز را بشناسند. یکی از مزایای آنها میتواند تعیین ساختار یک برنامه باشد. با این وجود، ساخت زیر-درخت ها کار آسانی نیست.	درخت های تجزیه
برنامه نویسی دقیق این روش اطمینان حاصل میکند که زمان محاسباتی در محدوده مناسب قرار می گیرد. با ای حال این روش دیگر مورد پشتیبانی فعال سازندگان نیست. درجه یا شاخص شباهت مطرح شده بین ۰ تا ۱ ، بدلیل ناتوانی روش برنامه نویسی دینامیک برای کنترل اجزای کد ترانهاده، با مشکل مواجه میشود.	SIM

جدول ۸ : مزایا و معایب روش های تشخیص سرقت ادبی در کد نویسی

فصل ششم : نتیجه گیری و تحقیقات آینده

نتیجه گیری

سرقت ادبی عبارت است از سوء استفاده، انتشار، اقتباس یا جایگزینی آثار، عقاید، و تفکرات دیگران بنام خود و عدم رعایت مالکیت معنوی آنها. برای جلوگیری از فرایند سرقت ادبی، تنها یادآوری این مسئله به دانشجویان که سرقت ادبی کار درستی نیست، کافی نخواهد بود. فرایند شناسایی سرقت ادبی کاری است که باید برای حداقل سازی این عمل نادرست انجام شود.

با افزایش استفاده از اسناد دیجیتال و اینترنت، سرقت ادبی نیز در حال رشد است. از این دیدگاه، ما همچنین شاهد رشد روش‌های تشخیص سرقت ادبی بوده‌ایم و پیشرفت‌های زیادی را در زمینه تشخیص اتوماتیک سرقت ادبی مشاهده کرده‌ایم. روش‌های گوناگونی برای تشخیص سرقت ادبی در حیطه‌های تک زبانی و چند زبانی، روش‌های مبتنی بر گراف، روش‌های معنایی، روش‌های برداری، روش‌های آماری، و غیره ارائه شده‌اند.

تحقیقات آینده

ابزارهای کنونی مقابله با سرقت زدگی برای سازمان‌های آموزشی، ناشران و افراد آکادمیک عموماً می‌توانند سرقت ادبی کلمه به کلمه را تشخیص دهند. در حقیقت سرقت ایده‌ها بسیار رایج تر از انواع دیگر سرقت‌های عددی است زیرا افراد آکادمیک زمان کافی را برای کشف ایده‌های جدید ندارند و ناشران ممکن است

ابزارهای کافی را برای تشخیص سرقت ادبی انجام شده ندارند. با پیچیده‌تر شدن انواع سرقت‌های ادبی، به

نظر می‌رسد که لازم باشد ابزارهای موجود در جهت کشف سرقت‌های ادبی، به حوزه سرقت‌های ادبی

معنایی، ساختاری و محتوایی بیشتر بپردازند و یا آنکه ابزارهای جدیدی در این راستا پدید آیند.

١. El Tahir Ali, A., et al. *Using Kohonen Maps and Singular Value Decomposition for Plagiarism Detection*. in *Computational Intelligence, Communication Systems and .. IEEE ٢٠١١ Third International Conference on. ٢٠١١ Networks (CICSyN)*, ..
٢. D. Abdulla, and V. Snásel. *Overview and Comparison of Plagiarism*. Ali, A.M.E.T., H.M. ٢٠١١ *Detection Tools*. in *DATESO*.
٣. Stamatatos, E. *Plagiarism detection based on structural information*. in *Proceedings of the ٢٠١١ th ACM international conference on Information and knowledge management*. ٢٠١١ .ACM
٤. Osman, A.H., et al., *An improved plagiarism detection scheme based on semantic role labeling*. ١٥٠٢-١٤٩٣): p. ٥(١٢. ٢٠١٢ *Applied Soft Computing*, Ceglarek, D. and K. Haniewicz. *Fast plagiarism detection by sentence hashing*. in *Artificial .. Springer ٢٠١٢ Intelligence and Soft Computing*.
٥. Sánchez-Vega, F., et al., *Determining and characterizing the reused text for plagiarism*. ١٨١٣-١٨٠٤): p. ٥(٤٠. ٢٠١٣ *Expert Systems With Applications*, and P. Rosso, *Methods for cross-language plagiarism* ,Barrón-Cedeño, A., P. Gupta ٢٠١٧-٢١١): p. ٥٠. ٢٠١٣ *Knowledge-Based Systems*, Potthast, M., et al., *Cross-language plagiarism detection*. *Language resources and .. ٦٢-٤٥): p. ١(٤٥. ٢٠١١ evaluation, based Natural Language Processing: Graph edit distance applied to -Røkenes, H.D., Graph ٢٠١٢ the task of detecting plagiarism*.
٦. Bao, J.P., C. Lyon, and P.C. Lane, *Copy detection in Chinese documents using Ferret*. ٣٦٥-٣٥٧): p. ٤-٣(٤٠. ٢٠٠٦ *Language resources and evaluation*, Khan, M.A., et al. *Copy detection in urdu language documents using n-grams model*. in *International ٢٠١١ Computer Networks and Information Technology (ICCNIT)*, .. IEEE ٢٠١١ *Conference on*.
٧. Das, S.N., M. Mathew, and P.K. Vijayaraghavan. *An Efficient Approach for Finding Near Duplicate Web pages using Minimum Weight Overlapping Method*. in *Information ٢٠١٢ Ninth International Conference on. ٢٠١٢ Technology: New Generations (ITNG)*, .. IEEE
٨. Cosma, G. and M. Joy, *An approach to source-code plagiarism detection and investigation* -٣٧٩): p. ٣(٦١. ٢٠١٢ *using latent semantic analysis*. *Computers, IEEE Transactions on*, ٣٩٤
٩. Tresnawati, D. and R. Syaichu, *Plagiarism Detection System Design for Programming Assignment in Virtual Classroom based on Moodle*. *Procedia-Social and Behavioral ١٢٢-١١٤): p. ٦٧. ٢٠١٢ Sciences*,
١٠. Citeseer ٢٠٠٧ Roy, C.K. and J.R. Cordy, *A survey on software clone detection research*, : ٢٣. ٢٠٠٥ Ellis, M.G. and C.W. Anderson, *Plagiarism Detection in Computer Code*. Mar, ١٠-١ p.
١١. Hage, J., P. Rademaker, and N. Vugt, *A comparison of plagiarism detection tools*. Utrecht ٢٠١٠ University. Utrecht, The Netherlands, ٢٨: p. ٢٠١٠

