Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3599-3611
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6140

Computer-based plagiarism detection techniques: A comparative study

Marwah Najm Mansoor[a], Mohammed S. H. Al-Tamimi[b,*]

[a]*Research and Development Department, Ministry of Higher Education and Scientific Research, Iraq*
[b]*Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Plagiarism is becoming more of a problem in academics. It's made worse by the ease with which a wide range of resources can be found on the internet, as well as the ease with which they can be copied and pasted. It is academic theft since the perpetrator has "taken" and presented the work of others as his or her own. Manual detection of plagiarism by a human being is difficult, imprecise, and time-consuming because it is difficult for anyone to compare their work to current data. Plagiarism is a big problem in higher education, and it can happen on any topic. Plagiarism detection has been studied in many scientific articles, and methods for recognition have been created utilizing the Plagiarism analysis, Authorship identification, and Near-duplicate detection (PAN) Dataset 2009-2011. Verbatim plagiarism, according to the researchers, plagiarism is simply copying and pasting. They then moved on to smart plagiarism, which is more challenging to spot since it might include text change, taking ideas from other academics, and translation into a more difficult-to-manage language. Other studies have found that plagiarism can obscure the scientific content of publications by swapping words, removing or adding material, or reordering or changing the original articles. This article discusses the comparative study of plagiarism detection techniques.

Keywords: Plagiarism, Academic, Detection, Dataset, Pan

1. Introduction

Plagiarism is a complicated and ethically difficult subject that refers to the act of stealing and publishing another author's work under one's name without crediting the original author. [20] Plagiarism is a type of deception. To adhere to ethical standards, authors must adequately credit their

*Corresponding author
Email addresses: mar118wa@gmail.com (Marwah Najm Mansoor), m_altamimi75@yahoo.com (Mohammed S. H. Al-Tamimi)

Received: November 2021 *Accepted:* January 2022

sources, and plagiarism breaches this obligation. Occasionally, though, the writers' pupils would fail to cite their sources properly. These difficulties are primarily the result of a shortage of information about correct citation use. As a result, plagiarism should be avoided to maintain ethical standards [11]. Perhaps the most appropriate description of plagiarism is "inadvertent copying of written material or computer code."[27]. As a result, one must be determined in their opposition. On the other hand, plagiarism is a widespread problem that impacts virtually every business. While plagiarism might occur accidentally, it is more often than not the outcome of a planned method [14]. Plagiarism has been more prevalent in recent years due to the amount of material available on the World Wide Web throughout the digital era (WWW). The use of statistical or automated approaches to identify plagiarism in natural languages began in the 1990s, with research on copy detection mechanisms in digital texts acting as a forerunner [15]. Since the 1970s, researchers have been examining computer code plagiarism in the Pascal and C programming languages to filter out code clones and software abuse [23]. To combat plagiarism, an enormous amount of research has been invested in software detection systems over decades [19, 7]. Initially, plagiarism was detected manually (by hand) or by comparing the text to previously examined material. Manual detection has become increasingly challenging in the modern day due to the amount of freely available online content. As a result, it is crucial to build automatic plagiarism detectors [18].

2. Plagiarism Type

Textual plagiarism and source code plagiarism are the two forms of PD methods; as shown in Figure [**?**], different types of PD approach[28]

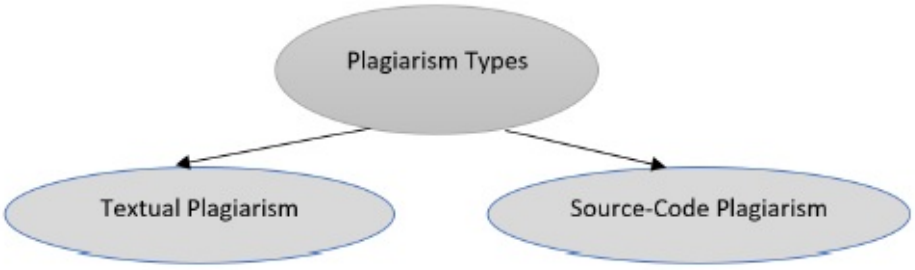

Figure 1: Plagiarism Type [11]

2.1. Textual Plagiarism

This type of plagiarism is the most common in research and scientific fields. The entire text or document is taken without referring to the author or mentioning a quotation. It can be further subdivided into seven sub-classes, as described in the sub-classes of textual plagiarism and Sub-Classes Textual Plagiarism is[12, 8]:

- **Copy-Paste Plagiarism:** This process involves copying the original text as if it were your work, without acknowledging the authors or the original paper.
- **Paraphrasing Plagiarism**: It is classified into two categories**:**

  (i) **Simple Paraphrasing:** The original text is presented differently by replacing the words with similar ones with the same meaning.

(ii) **Mosaic / Hybrid / Patchwork Paraphrasing:** The text results from combining different contributions from different papers and presented differently without referring to the original citation of the works.

- Metaphor Plagiarism: Presenting other ideas in better ways.
- Idea Plagiarism: The entire solution and ideas are stolen from others, claiming that it is an original research paper.
- Recycled Plagiarism: The authors here use their previous/old works and papers for a new publication.
- 404 Error / Illegitimate Source Plagiarism: When the citation of the works is invalid.
- Re-Tweet Plagiarism: In this type, the citation is referred to, but there is no difference between the original work and the author's work from the point of structure, grammar, and words.

2.2. Source-Code Plagiarism

It appears typically in educational fields, where the programming code of a specific program is written originally by someone, and it is copied, adjusted, or reused by others partially or completely. It has five categories and is discussed below[12, 24]:

- Strings: While this approach will match the strings, it is possible to conceal this plagiarism by changing the source code identifiers.
- Tokens: As a first step, a lexer converts the programmer into a token. Codes with identifiers, whitespaces, or comments will be ignored.
- Parse Trees: Both source codes are assigned here. These trees are then compared. It is said that the source codes are similar if both trees are equal, but otherwise, they are not.
- Program Dependency Graphs (PDGs): The actual flow of control can be captured using PDGs. These PDGs can detect equivalence, but they need a lot of work and are difficult to use.
- Metrics: It assigns numerical values to code segments based on specified parameters. The score mentioned above is determined by the number of loops, conditional statements, and variables in the code. Textual and source code plagiarism can be detected by humans or automated detection methods.

3. Langue Type

Plagiarism can be divided into two basic categories[16, 20]:

1. **Monolingual.**
2. **Cross-lingual.**

PD can be classified based on the language of the texts being processed as **(Mono-Lingua)** if the source and suspect documents use the same language or (Cross-Lingual) if the languages are diverse. Automatic PD uses a reference corpus that compares a suspect text to a collection of papers to identify the source of the plagiarized pieces. The source and suspect documents may be written in the same language (Mono-Lingua) such as English-English or different languages (Cross-Lingual), plagiarism categories shown in Figure 2

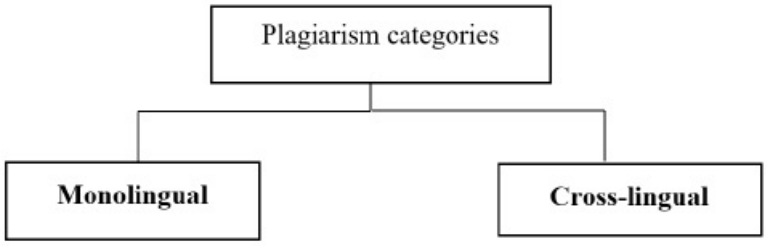

Figure 2: Plagiarism categories

4. Plagiarism detection

Plagiarism detection techniques are essential for identifying instances of plagiarism; the stolen material must be distinguished from the original by a plagiarism detection function. This procedure can occasionally validate the quantity of material that is plagiarized [18]. PD is the method of separating the document's characteristics, assessing its content, identifying potentially plagiarized sections, and getting similar remaining documents to light if they are accessible. This method can improve PD performance by eliminating the selection of source texts and incorporating semantic relationships between words and their structural composition [10]. Sentences with a high degree of resemblance to suspicious text sentences but distinct meaning Plagiarism detection system is shown in Figure 3[4]

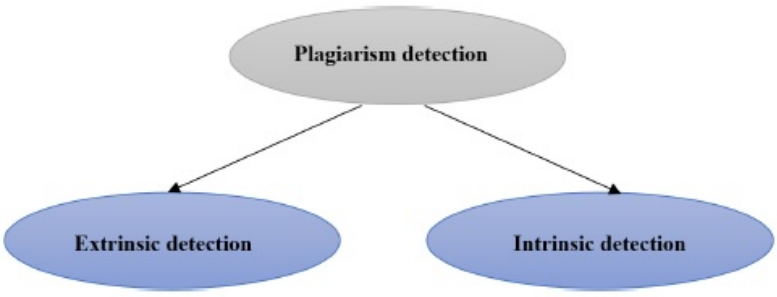

Figure 3: Plagiarism detection

- Extrinsic detection: A suspected document is examined against a reference source document corpus or collection in an extrinsic PD system, PDS. This reference collection can be online or offline, consisting of online sources on the World Wide Web or an offline database containing the source materials[26].
- Intrinsic detection: Intrinsic PD, authorship verification, authorship attribution, and authorship attribution are related but distinct activities. Each quantifies the author's writing style and/or analyzes the text's characteristics and complexity.[11]

5. Plagiarism comparative study

Researchers have implemented many methods to overcome Plagiarism as it has grown to Form a serious issue among the academic community; researchers have used different methods to overcome these activities [11]. Therefore, a comparative study about Plagiarism detection as viewed by researchers will be illustrated in Table 1.

Table 1: Comparative study

Refers	method	Illustration	limitation
[25]	Detection of Plagiarism Using a Trie-tree Data Structure	For high-speed data comparison, both character-based and knowledge-based methods are utilized.	A comparison-based method necessitates content processing, which is inefficient for large files.
[17]	Rabin Karp Method Vivek	Utilized a sampling module to reduce the size of the dataset and a cost function to detect document repetition; calculated both syntactic and semantic similarity for document similarity detection.	The approaches are limited at one side of matching but lacking in producing the refined result with high throughput and similarity
[29]	A Code Plagiarism Detection Algorithm Based on the AST	They proposed the AST-CC method for generating and comparing hash values.	Reduced efficiency in storing data structures
[3]	A Review of the State of the Art in Source Code Plagiarism	Numerous techniques, such as natural language processing and machine learning	It is challenging to identify plagiarism across many source codes for various languages.
[22]	"Longest common consecutive word"	The outsourcing suffix tree method is a numerically based comparison technique.	The disadvantage of this suggested method is its lengthy loading time.
[13]	Methods for detecting clones and assessing similarity.	It has an anti-plagiarism technology that allows for the management of massive datasets.	It is necessary to handle many papers with a similar identity and a high execution rate.
[1]	PDLK: Plagiarism detection using linguistic knowledge	The suggested approach enlarges the words in phrases to address the issue of limited information.	It used only 200 documents from 22000 documents that do not work on all datasets, and the result only evolves on this document.

[21]	A linguistic Treatment	The proposed approach can identify several types of plagiarism, such as exact verbatim copying, paraphrasing, sentence transformation, and word structure alteration.	The system does not contain a thorough weighting of variants of the linguistic feature functions to allow for a more in-depth examination of the system.
[2]	Semantic Role Labeling (SRL)	The proposed system presents an External PD System (EPDS)	Because there are 22000 English books and the system does not inspect the impact of stop words on the text's significance, the number of documents used in testing and training is limited.
[5]	Similarity system by using dice	The suggested system employs an extrinsic plagiarism detection technique inspired by cognition since it uses semantic information to detect copied material without requiring human intervention.	This method is incapable of detecting plagiarism in the captions of figures, figures, or flowcharts. It is restricted to a text format, and no machine learning techniques are used to identify plagiarism in the document content.
[6]	Similarity system using hybrid technology	The suggested approach included techniques from "natural language processing (NLP)" and "machine learning (ML)," as well as an external strategy for identifying plagiarism that made use of text mining and similarity analysis. The suggested method combines Jaccard and cosine similarity.	It is possible to use another dictionary and increase the number of stop words

The following is an explanation of the previous studies mentioned in the above table: In [25], the author **(Talebpour et al.)** provides a proposed method for comparing legitimate and suspect text documents. They illustrate their point with Persian PlagDet text papers. Both character-based and knowledge-based detection approaches have aided in improving our system. Additionally, it is a method for insertion and retrieval that has enabled rapid comparison of lengthy documents. The System constraints are in A comparison-based technique that requires content processing, which is inefficient for large files. And in [17] **(v. Kumar et al.)** Compressive sensing-based Rabin Karp (CS-RKP) is presented as an advanced new method. This technique utilized a sampling module to reduce the size of the dataset and a cost function to discover document repetition, calcu-

lating both syntactic and semantic similarity throughout the content. The result assertion evaluates computation time; the similarity measure w.r.t. n-grams for various values of N demonstrates the proposed algorithm's efficiency, and the system limitations are that the approaches are limited on the one hand in terms of matching but fall short on the other in terms of producing refined results with high throughput and similarity. And [29] (**J. Zhao et al**.) They provide a more effective method for detecting plagiarism that is based on an "abstract syntax tree (AST)" by computing and comparing the "hash values of the syntax tree nodes." To guarantee appropriate implementation of the technique, special attention must be taken to reduce error rates while computing the hash values of operations, particularly mathematical operations such as subtraction and division. The test results established that the measurement is both reliable and essential. It performs admirably in code comparison and is beneficial in the area of copyright protection for source code. The system's shortcomings are Reduced efficiency in storing data structures [3]. the author **(M.Agrawal, and . and et al.)** presented several approaches and algorithms for detecting source code plagiarism. Thus, utilizing these approaches, a company or academic institution may easily detect plagiarism in the source code. Distinguish between these many plagiarism methods to determine how one strategy interacts with the others and the restriction that it is difficult to identify plagiarism among various source codes written in various languages. And in [22], the author (A. Sediyono, and . and et al.) presents a numerical comparison method similar to computing time without sacrificing the word order of common components. According to the experiment, the suggested method outperforms the suffix tree for observed paragraphs with less than one hundred words in length. The constraints It is necessary to handle many papers with a similar identity and a high execution rate. And in [13], the author **(M. Ďuračík. et al.)** evaluate state of the art in the field of source code analysis, with a focus on plagiarism detection, and make recommendations for future work in this area. Plagiarism detection techniques include those for detecting clones and determining similarity. It may be classified into three categories. The first application is text-based, taking the only plain text as input. The second level is token-based. The top-level is model-based, and models are used to represent source code. Due to the inability of these complex algorithms (token and model-based) to process large datasets, it is required to examine many documents with similar identities that exhibit excellent performance. think that algorithms capable of handling massive source code are the wave of the future. These algorithms should be fundamentally model-based. They may be used to develop large-scale anti-plagiarism systems. Additionally, they may be used to optimize source code. And in [1], the author **(Abdi et al**.) presents an approach for identifying external plagiarism that exploits the semantic links between phrases and their syntactic structure. The problem with present techniques is that they do not adequately capture the substance of a contrast between a source document sentence and a suspicious document sentence when the two sentences contain the same surface text (the words are identical) or are paraphrases of one another. The constraint is It was used on 200 papers out of 22000 documents and did not work on all datasets, resulting in only one document evolving. And in [21], the author **(M. Sahi . and et al.)** The suggested approach to detecting plagiarism combines semantic and syntactic similarities between text pieces. This innovative technique utilizes non-linear linguistic information sources by utilizing a lexical database to determine the relatedness of text texts. The suggested technique performs cognitive-inspired computing by using semantic knowledge. The framework can identify intelligent instances of plagiarism, such as verbatim copying, paraphrasing, sentence rewording, and sentence transformation. The system does not contain a thorough weighting of variants of the linguistic feature functions to allow for a more in-depth examination of the system. In [2], the author **(A. Abd . and et al.)** The proposed approach avoids choosing source text phrases that are close to suspect text sentences yet have a separate meaning. On the other hand, an author may transform an active sentence into a passive sentence and vice versa; hence, the technique

integrated the SRL methodology to handle the issue mentioned above. Additionally, the technique used a content word expansion approach to fill lexical gaps and uncover concepts presented in novel ways. The suggested model can identify several types of plagiarism, such as exact verbatim copying, paraphrasing, sentence transformation, and word structure alteration. The testing findings indicate that the proposed approach can increase performance compared to the PAN-PC-11 participating systems and other currently used methodologies. Because there are 22,000 English novels and the algorithm does not consider the effect of stop words on the text's significance, the number of documents used for testing and training is limited. In [5] the author**(L. Ahuja and et al.)** The proposed technique use the Dice measurement as a similarity metric to assess the semantic similarity of two sentences. Additionally, it makes use of linguistic characteristics such as path similarity and depth estimation to estimate the similarity of two words, and these features are weighted differently. It is capable of identifying rewriting, paraphrasing, verbatim copying, and plagiarism through the use of synonyms. It was evaluated using the PAN-PC-11 corpus, and its limitation is that it cannot detect plagiarism in figure captions, figures, or flowcharts because it is limited to a text format and no machine learning approaches are used to detect plagiarism in document content. However, the result is not significantly different from the previous study. And in [6] The authors k. Farah and s. Mohammed proposed a system that integrated "natural language processing (NLP)" and "machine learning (ML)" approaches, as well as an external strategy for detecting plagiarism based on text mining and similarity analysis. The proposed approach makes use of a combination of Jaccard and cosine similarity. It was evaluated using the PAN-PC-11 corpus, and the suggested approach is backed up by a design application for detecting plagiarism in scientific papers and generating reports in a non-modifiable format such as "Portable Document Format (PDF)" and the system's restriction are that another dictionary can be used to enhance the number of stop words.

Plagiarism Method

Includes studies on the automatic detection of possible cases of plagiarism. Papers that utilize lexical, syntactic, and semantic similarity analysis and similarity of non-textual content components such as citations, images, tables, and mathematical formulae. That evaluate plagiarism detection strategies, for example, by giving test sets and reporting on performance comparisons, and the most effective technique used in PD is shown in Table 2.[20, 4]:

Table 2: Plagiarism Method's

Methods	**Language**	**classification**	**Type**
Character-Based	**Mono-lingual**	**literal**	**External**
Vector-Based	**Mono-lingual**	**literal**	**External**
Syntax-Based	**Mono-lingual**	**literal**	**External**
Semantic-Based	**Mono-lingual**	**Literal/intelligent**	**External**
Fuzzy-Based	**Mono-lingual**	**Literal/intelligent**	**External**
Structural-Based	**Mono-lingual**	**literal**	**External**
Stylometric-Based	**Mono-lingual**	**literal**	**Internal**
Cross-Lingual	**Cross-Lingual**	**literal**	**External**
Grammar-Based	**Mono-lingual**	**literal**	**External**
Classification &Cluster-Based	**Mono-lingual**	**literal**	**External**
Citation-Based	**Mono-lingual**	**Literal/ intelligent**	**External**

6. Plagiarism Tool

With the use of plagiarism detection technologies, a researcher can ascertain whether another individual plagiarized his/her study paper. It aids in the development of writing abilities, as specific plagiarism programs also check for grammar. Plagiarism detection software enables access to numerous databases. It raises awareness of plagiarism among researchers and faculty members and assists them in developing successful academic careers in the future by avoiding plagiarism:[4, 9, 11]

- MOSS: 1994 MOSS (Measure of Software Similarity). This detects source code plagiarism; it takes parts of the code as an input and produces HTML pages as an output to analyze the similarities between pairs of documents. Its open-source, Enables the exclusion of templates, Can be used for 25 programming languages. However, online use only has difficulty with various forms of plagiarism since it makes a concerted effort to prevent false positives, therefore rejecting a large amount of information.

- Authenticate 1996 is a text-document-based plagiarism detection tool presented as a web page. It compares the number of documents with the original one without installing on the end-user computer, but it is limited to 25,000 words per time.

- JPlag :1997 this type is an online source-code plagiarism tool. It takes a number of programming codes and selects identical lines. It works with C, C++, and Java programming languages to detect hundreds of code lines in less than one minute. Due to the fact that it parses the results, it is possible for there to be no scores if the source code has a single tiny error.

- GPSP - Glatt Plagiarism Screening Program: 1999 Unlike the previous tools, it works offline, and this tool mix different approaches and finds the similarities among the writing styles of differed authors to reveal Plagiarism by making the author goes through a fill-in-the-blank test, then it counts the correctly filled blanks and the time is taken to finish the test, finally according to the results it decides an act of plagiarism.

- Turnitin: 2000 IParadigms provide it as a web-based tool. The user must upload his/her required document online, and then the document will be saved to the system's database. It accepts nearly 15,000 Institutions around the works with more than 30 million users for its flexibility and robustness. Therefore, it is considered the best tool.

- Plagiarism Checker: 2006 is a free and online tool, using search engine services to detect students' plagiarism by checking their documents if it contains a similar copy from another online document.

- Plagiarism Scanner: 2008 It is an effective tool that detects online. When detection for plagiarism is found, it produces a full report, including the rate, originality, and percentage of plagiarized materials.

- Plag tracker: 2011 accommodates a large number of academic resources in its database system and produces a detailed report whenever plagiarism is detected.

- PlagScan: 2015 provides multiple services to companies, universities, and schools, but it is not free, and the users must have a paid account to register.

- Exactus Like: 2016 is a web-based online tool that works with different formats like HTML. A deep parsing function detects moderately disguised borrowing (word/phrase reordering, substituting some words with synonyms).
- Grammarly: 2016 is a website and a mobile application service that offers an excellent opportunity to the individuals to correct their documents within a real-time manner and a friendly user interface, and it works online.
- Grammarly: 2018 It is an evolved version of the previous one. It is the premium type. It is targeted business industries such as teams and companies.
- Dupli Checker: 2020 It is one of the most effective-free plagiarism tools on the internet. The user only requires a search engine and a connection to the world wide web to access this tool. It enables the user to either copy-paste or upload the document to check for plagiarism.

And the following table summarizes what was mentioned above:

Table 3: plagiarism tool

Number	Name	Year	Description	Type
1	MOSS	1994	detection for source code plagiarism	Online
2	Ithenticate	1996	detection for text-document	Online
3	JPlag	1997	detection for source-code plagiarism tool	Online
4	GPSP	1999	detection for text-document	Off-line
5	Turnitin	2000	detection for text-document	Online
6	Plagiarism Checker	2006	detection for text-document	Online
7	Plagiarism Scanner	2008	detection for text-document	Online
8	Plag traker	2011	detection for text-document	Off-line
9	PlagScan	2015	detection for text-document	Online
10	Exactus Like	2016	detection for text-document	Online
11	Grammerly	2016	detection for text-document	Online
12	Grammerly	2018	detection for text-document	Online
13	Dupli Checker	2021	detection for text-document	Online

7. Dataset

PAN Dataset is a famous dataset used to evaluate plagiarism detection algorithm.[16] , PAN data set was created from 2009 to 2020, and beast addition to evaluation is PAN2011; it means plagiarism has been inserted into the documents contained in this corpus, both manually and automatically. It is assumed that these documents will help with evaluating automatic PD algorithms.[30]:

- ***Source Documents***: This corpus contains documents derived from Project Gutenberg books (www.gutenberg.org). The total collection has 22,000 English books.
- ***The license of the Corpus:*** All of the texts in this corpus are in the public domain, to the best of our knowledge. As a result, the corpus can be used without cost or legal consequences.

- ***Plagiarism in the Corpus:*** There are no actual instances of plagiarism here. Instead, the annotated plagiarism cases are simulated, meaning they were created by a human, to confirm that they are not claiming that any of the authors included in this corpus were plagiarized in real life. Plagiarism can also be artificial, meaning a computer programmer creates it. As a result, any resemblance to accurate or actual plagiarism is purely coincidental. And the following table shows the description of three fames tyble 3 datasets used to evaluate the algorithm:

Table 4: Type of PAN dataset

Name	Year	Size	Type	Description
PAN -PC-9	2009	2GB	Document	The PAN plagiarism corpus 2009 (PAN-PC-09) is used to test plagiarism detection systems automatically. The corpus can be utilized for free for research purposes. The PAN-PC-09 comprises materials that have been automatically inserted with false plagiarism. The plagiarism cases were created using computer software known as a random plagiarist, which makes plagiarism based on a set of random variables. The plagiarism percentage throughout the entire corpus, the percentage of plagiarism per document, the length of a single plagiarized piece, and the degree of obfuscation per plagiarized section are among the factors. (as well as the fact that this version is out of date).
PAN- PC-10	2010	2GB	Document	The PAN plagiarism corpus 2010 (PAN-PC-10) is a corpus for evaluating automatic plagiarism detection algorithms. For research purposes, the corpus can be used free of charge. The PAN-PC-10 contains documents in which artificial plagiarism has been inserted automatically and documents in which simulated plagiarism has been inserted manually. The former has been constructed using a so-called random plagiarist. According to several parameters, this computer program creates plagiarism, while the latter has been obtained with crowdsourcing via Amazon's Mechanical Turk. (and this version is outdated)
PAN-PC-11	2011	2GB	Document	The PAN plagiarism corpus 2011 (PAN-PC-11) is used to test plagiarism detection systems automatically. The corpus can be utilized for free for research purposes. The PAN-PC-11 comprises documents that have been automatically plagiarized and documents that have been manually plagiarized. The former was created using a so-called random plagiarist. This computer program generates plagiarism based on a set of parameters, while the latter was created via crowdsourcing through Amazon's Mechanical Turk.

8. Conclusion

A detailed literature review of plagiarism kinds, strategies, and tools was conducted in this research. Text plagiarism, which has seven subtypes and source-code plagiarism, are two of the most common types of plagiarism. Then, over roughly 24 years, plagiarism detection methods and tools were demonstrated, with the newly produced tools being more assertive. Most of the tools function online with an internet connection and a web page, with some of them being free and others requiring a subscription fee. The most well-known dataset (PAN) used in plagiarism detection was then examined. Finally, a discussion of the most common types of plagiarism was conducted and the most challenging aspects of deploying plagiarism detectors.

References

[1] A. Abdi, N. Idris, R.M. Alguliyev and R.M. Aliguliyev, *PDLK: Plagiarism detection using linguistic knowledge*, Expert Syst. Appl. 42(22) (2015) 8936–8946.

[2] A. Abdi, S.M. Shamsuddin, N. Idris, R.M. Alguliyev and R.M. Aliguliyev, *A linguistic treatment for automatic external plagiarism detection*, Knowledge-Based Syst. 135 (2017) 135–146.

[3] M. Agrawal and D.K. Sharma, *A state of art on source code plagiarism detection*, Int. Conf. Next Gener. Comput. Technol. NGCT, 2016, pp. 236–241.

[4] R.A. Ahmed, *Overview of Different Plagiarism Detection Tools*, Int. J. Futurist. Trends Engin. Technol. 2(10) (2015) 2–4.

[5] L. Ahuja, V. Gupta and R. Kumar, *A new hybrid technique for detection of plagiarism from text documents*, Arab. J. Sci. Eng. 45(12) (2020) 9939–9952.

[6] F.K. Al-Jibory and M.S.H.A. Tamimi, *Hybrid system for plagiarism detection on a scientific paper*, Turk. J. Comput. Math. Educ. 12(13) (2021) 5707–5719.

[7] E.S. Al-Shamery and H.Q. Gheni, *Plagiarism detection using semantic analysis*, Indian J. Sci. Technol. 9(1) (2016) 1–8.

[8] S.M. Alzahrani, N. Salim and A. Abraham, *Understanding plagiarism linguistic patterns, textual features, and detection methods*, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42(2) (2012) 133–149.

[9] S. Awasthi, *Plagiarism and academic misconduct: A systematic review*, DESIDOC J. Libr. Inf. Technol. 39(2) (2019) 94–100.

[10] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro and M.D. Esposti, *A plagiarism detection procedure in three steps: Selection, matches and squares*, CEUR Workshop Proc. 502 (2009) 19–23.

[11] A.S. Bin-Habtoor and M.A. Zaher, *A survey on plagiarism detection systems*, Int. J. Comput. Theory Eng. 10(8) (2012) 185–188.

[12] H.A. Chowdhury and D.K. Bhattacharyya, *Plagiarism: Taxonomy, tools and detection techniques*, arXiv preprint arXiv:1801.06323, 2018.

[13] M. Ďuračík, E. Kršák and P. Hrkút, *Current trends in source code analysis, plagiarism detection and issues of analysis big datasets*, Procedia Eng. 192 (2017) 136–141.

[14] D. Gañan, *Plagiarism Detection*, Lect. Notes Data Eng. Commun. Technol., 34(2020) 19–40.

[15] J. Kasprzak, M. Brandejs and M. Křipač, *Finding plagiarism By evaluating Document similarities*, CEUR Workshop Proc. 502 (2009) 24–28.

[16] P. Gupta, K. Singhal, P. Majumder and P. Rosso, *Detection of paraphrastic cases of mono-lingual and cross-lingual plagiarism*, IR-Lab,DA-IICT, India, (2011) 1–6.

[17] V. Kumar, C. Bhatt and V. Namdeo, *A framework for document plagiarism detection using Rabin Karp method*, Int. J. Innov. Res. Technol. Manag. 3404(4) (2021) 17–30.

[18] S. Prasanth, R. Rajshree and B.S. Balaji. , *A Survey on plagiarism detection*, Int. J. Comput. Appl. 86(19) (2014) 21–23.

[19] M. Potthast, B. Stein, A. Barrón-Cedeño and P. Rosso, *An evaluation framework for plagiarism detection*, Coling 2010 - 23rd Int. Conf. Comput. Linguist. Proc. Conf., 2010, pp. 997–1005.

[20] A.H. Osman, N. Salim and A. Abuobieda, *Survey of text plagiarism detection*, Comput. Eng. Appl. J. 1(1) (2012) 37–45.

[21] M. Sahi and V. Gupta, *A novel technique for detecting plagiarism in documents exploiting information sources*, Cognit. Comput. 9(6) (2017) 852–867.

[22] A. Sediyono, K. Ruhana and K. Mahamud, *Algorithm of the longest commonly consecutive word for plagiarism detection in text based document*, 3rd Int. Conf. Digit. Inf. Manag. ICDIM, 2008 pp. 253–259.
[23] A. Sharma, V. Walia and M. Gahlawat, *Review: Plagiarism an act of unethics*, PharmaTutor Mag. 3(2) (2016) 20–23.
[24] D. Sraka and B. Kaučič, *Source code plagiarism*, Proc. Int. Conf. Inf. Technol. Interfaces, ITI, no. July, 2009 (2009) 461–466.
[25] A. Talebpour, M. Shirzadi Laskoukelayeh and Zahra Aminolroaya *Plagiarism detection based on a novel trie-based approach*, Forum Inf. Retriev. Eval. (2016) 109–117.
[26] K. Vani and D. Gupta, *Study on extrinsic text plagiarism detection techniques and tools*, J. Eng. Sci. Technol. Rev. 9(5) (2016) 9–23.
[27] K. Vani and D. Gupta, *Text plagiarism classification using syntax based linguistic features*, Expert Syst. Appl. 88 (2017) 448–464.
[28] K. Vani and D. Gupta, *Unmasking text plagiarism using syntactic-semantic based natural language processing techniques: Comparisons, analysis and challenges*, Inf. Process. Manag. 54(3) (2018) 408–43.
[29] J. Zhao, K. Xia, Y. Fu and B. Cui, *An AST-based code plagiarism detection algorithm*, Proc. 10th Int. Conf. Broadband Wirel. Comput. Commun. Appl. BWCCA, 2015, pp. 178–182.
[30] Https://pan.webis.de/, *No Title*, Https://pan.webis.de, 2011.