High Perfomance Plagiarism Detection Using Rabin's fingerprint and Adaptive N-gram Methodologies

Karuma Chege, Dr George Okeyo PhD, Dr Richard Rimiru PhD

Abstract— Plagiarism poses a grave danger to the integrity of the higher education process. To address this threat, academic institutions invest in computerized tools that assist in verifying that academic works submitted by students and researchers are their original work. The available tools largely employ a combination of text fingerprinting and n-gram techniques to identify similarity in content between submitted documents and reference indices they maintain. For the fingerprinting requirement, MD5 and SHA-1 were found to be the most widely used schemes. This research explored the use of the Rabin's fingerprinting scheme as a replacement for MD5 and SHA-1.Rabin's fingerprint was selected due to its weaker cryptographic profile as compared to MD-5 and SHA-1 which makes it less computationally intensive. The research also sought to determine the size of n-grams that would be most effective for use in n-gram based plagiarism detection tool. Using a prototype application, it was established that the Rabin's fingerprint outperforms MD-5 and SHA-1 by factors of 2.89 and 2.96 receptively. Results derived from the prototype also indicated the effective n-gram size as 4. Further, the research established that the addition of n-gram rolling added to the overall plagiarism detection effectiveness by a factor of 3.02.

Index Terms— Rabin's fingerprint, plagiarism, fingerprint, n-gram, MD5, SHA-1, stylometry.

1. INTRODUCTION

1.1 Background

The availability of opportunities for tertiary education continues to rise globally as governments and private enterprises accelerate their investments into it as established in [24]. Universities and other institutions of higher learning are continually trying to ensure that they produce graduates of the highest caliber as this has a positive effect on their ability to secure talented students as well as ensuring they are better placed to secure more funding for their programs. As such, students are continually under pressure to demonstrate that their research based undertakings and subsequent examinable outputs are conducted in the context of complete academic honesty. The growth of the internet has led to the universal availability of voluminous amounts of information covering every conceivable topic, this has only served to provide increased avenues for dishonest students and researchers to be able to easily incorporate other people's work into their own and subsequently attempt to represent the same as their original work.

1.2 What is plagiarism?

This is the deliberate incorporation of the ideas, thoughts and expressions of another party's work into someone's work and its subsequent presentation as their original work as derived from [25]. In the field of academia, plagiarism is considered a form of academic dishonesty and institutions address it by applying penalties such as suspension, expulsion or delayed graduation for the offending parties. In extreme cases, academic plagiarism has been the subject of court proceedings as seen in [10]. In the publishing and media industries, it's considered a violation of journalistic ethics and could lead to job loss, estrangement from the profession as well as legal proceedings in a court of law. The case of Jayson Blair as detailed in [19] is one of the most prominent incidences of plagiarism in journalism.

From the foregoing, plagiarism is a critical issue in fields of academia, software development, print media and entertainment with substantial efforts and resources being allocated to detecting and eliminating it. In the eras past, plagiarism detection was mainly a manual and laborious activity but in recent times, the availability of computerized tools has led to significant changes in the way it's done. Plagiarism checkers, as the tools are generally called, have greatly enhanced the art of plagiarism detection and their effectiveness is expected to continue on an increasing trajectory. Examples of such tools include TurnItIn, Viper and Plagium.

1.3 Definitions

According to [6], a text Fingerprint is a digest of some larger data item into a much shorter bit-string using a specified digesting scheme, and where the resulting bit-string is considered as the unique identity of the larger data item.

The Message Digest algorithm version 5 (MD5) is a cryptographic hash function that digests text fragments into 128-bit hash values which are usually rendered as hexadecimal numbers of 32 digits length as detailed in [7].

The Secure Hash Algorithm version 1 (SHA-1) is a cryptograph-

- *Karuma Chege is currently pursuing master's degree program in Computer Systems at the Jomo Kenyatta University of Agriculture and Technology (JKUAT), Kenya. E-mail: kchegeh@mail.com*
- *Dr George Okeyo is the Chairman, School of Computer Science at the Jomo Kenyatta University of Agriculture and Technology, Kenya.*
- *Dr Richard Rimiru is a lecturer at the School of Computer Science at Jomo Kenyatta University of Agriculture and Technology, Kenya.*

ic hash function that digests text fragments into 160-bit hash values which are usually rendered as hexadecimal numbers of 40 digits length as detailed in [29].

The Rabin's fingerprinting scheme is a method for deriving fingerprints of bit-strings that employs randomly chosen irreducible polynomials over a suitable finite field as defined in [21].

An N-gram, where n is an integer, is a construct employed in the fields of natural language processing and computational linguistics to describe a collection of n items which are contiguous with each other from some text or speech corpus as seen in [13].

An adaptive N-gram refers to the n-gram generation process where the n-grams are generated in a rolling manner. In this case the rolling window size is equal to the n-gram size in use.

1.4 Objective of the study

This research sought to demonstrate using a prototype application that a combination of Rabin's fingerprinting and adaptive n-grams can be used to develop an effective and highly efficient plagiarism detection tool.

1.4.1 Specific objectives

1. To research and determine if a suitable implementation of the Rabin's fingerprinting scheme provides better performance as compared to MD5 or SHA-1 for the plagiarism detection task.
2. To determine the n-gram size that provides the most optimal performance for the prototype while also maximizing the effectiveness of plagiarism detection.
3. Identify the impact on performance and effectiveness of the prototype when performing the n-gram generation using a rolling n-gram approach.

2 Related work

This study aimed at demonstrating that an effective plagiarism checking application could be created using the Rabin's fingerprinting scheme and adaptive n-gram techniques. Before this can be examined in detail, consideration is given to some of the approaches that have been used to address this problem previously and other related work.

2.1 String matching

In this approach, a target document is examined for verbatim text overlaps with a set of reference documents. Due to its computationally intensive nature, [12] proposed an approach where maximal matches in pairs of strings between the suspect document and the reference documents are obtained and are then used as plagiarism indicators.

This approach has been noted to be highly inefficient for use with large collections of reference documents as detailed [18].

2.2 Stylometric analysis

Stylometry is the application of statistical methods to identify an individual's linguistic style by examining the recurrence of particular expressions and the progressive development of ideas expressed in their works as defined in [9]. By examining various sections in a document using stylometric analysis, sections and chunks of the document which are not in conformity with the linguistic style of the rest of the document can be identified. Such instances can then be used as heuristic pointers to cases of plagiarism as argued by [16].

While stylometry is quite effective in identifying the inclusion of external work in an individual's work, it's greatly limited by the fact that on its own, it does not provide any way of establishing where the suspect content was derived from, as such; the accusation of plagiarism is not fully proven using this approach. Stylometry is also susceptible to manipulation and circumventing as demonstrated by [8]. In their research, they demonstrated adversarial stylometry where an author could conceal the discovery of his authoring style in order protect his identity.

2.3 Bag of words analysis

In this approach, multisets of constituent words are created from document sections and similarity analysis is then performed with multisets from documents in a reference collection as derived from [22]. For similarity analysis, two of the most frequently used methodologies are Standard Vector Space Models and Latent Semantic Analysis. A Vector Space Model is a representation of a text document as an algebraic model while Latent Semantic Analysis are techniques for analyzing relationships between text documents on the basis of the terms they contain [27]. In the Standard Vector Space Model approach, multisets are created from the target documents and the reference documents which are then represented as Vector Spaces Models and content similarity is determined on the basis of the Cosine Similarity Measure as explained in [23].

This approach is more suited to performing document classifications as it too simplistic for plagiarism detection especially when the reference collection is large [26].

2.4 Citation analysis

This is a relatively young approach to plagiarism detection that does not rely on the textual similarity between target documents and reference documents. It applies citation analysis to capture citation and referencing information in targeted documents with an aim of identifying similarities in content between target documents and the references they cite as derived from [11].

In [14] it's observed that this approach is mostly suited for use in highly scientific contexts and does not work very well for art, humanities and business related content.

2.5 Fingerprint analysis

This is the most prevalent plagiarism checking approach and it works by creating representative fingerprint digests of n-gram content in the target document and comparing it with a reference index which comprises of fingerprint digests of reference documents as detailed in [15]. The digests of the n-grams that maybe examined for similarity with the reference index are called minutiae. The fingerprinting approach provides wide latitude in the selection of the fingerprinting schemes that maybe applied. A fuzzy fingerprinting scheme that produces very similar fingerprints for closely related text content was proposed by [23] for use in text similarity determination. In their proposal, the Cosine Similarity Measure is derived after adapting fingerprints generated using their fuzzy fingerprinting scheme into a Vector Space Model and deciding on the basis of the achievement of a particular threshold that the texts were too similar than could be explained by random chance.

Shortcomings observed in fingerprinting based approaches towards similarity detection are mostly related to the fingerprinting schemes employed. In their research, [23] observed that MD5 is computationally expensive to perform making solutions based on it to be quite inefficient. They further observed that the Secure Hash Algorithms (SHA) family of fingerprinting schemes which includes SHA-1 are more adapted to cryptographic uses and their use outside of cryptography enhanced applications is likely to suffer from performance challenges. An additional consideration when using the fingerprinting approach is that fingerprints of reference documents will need to be stored, and as the reference index grows, it could require substantial storage space and the attendant tools to manage it.

2.6 The Rabin's fingerprinting scheme

The Rabin's fingerprinting scheme is a fingerprinting technique that employs randomly chosen irreducible polynomials over a suitable finite field to derive a numeric residue from a text document which is then designated as its unique fingerprint. Using this algorithm, APIs can be created that generate unique fingerprints for any digital text document in a very efficient manner. The resulting fingerprints can then be compared with other fingerprints generated using the same API for determining the equality of the documents. Documents that have same Rabin's fingerprint are duplicates of each other or are same document; likewise, fragments of digital documents that have the same Rabin's fingerprint have the same content. Implementations of the Rabin's fingerprint scheme are used in applications such as the Rabin–Karp string search algorithm as noted in [17] and in network file transfer management tools like The Low Bandwidth Network File system (LBFS) as documented in [20].

2.7 N-Grams

An n-gram is a construct that describes a collection of n items which are contiguous with each other derived from some text or speech corpus of interest. Depending on the context in which they are being used in, n-grams can be constructed based on letters, syllables and words. N-gram methodologies are extensively used in information retrieval, natural language modeling, speech recognition, protein sequencing and DNA sequencing applications. The critical importance of n-gram based techniques in natural language processing can be inferred from the observation that Google and Microsoft, two of the largest computer technology companies offer n-gram data for use by researchers through Google Books N-gram Viewer Services and Microsoft Web N-gram Services respectively. An overview of the applications of the Microsoft Web N-gram Services is given in [28].

2.8 Observed gaps in available knowledge

While reviewing the relevant literature regarding the use of Rabin's fingerprint and n-gram approaches in the creation of a plagiarism detection application, several gaps in the amount of available information were identified and are highlighted below.

1. **Suitability of the Rabin's fingerprinting scheme**
   While there exists research documentation that details the inefficiency of text fingerprint generation using MD5 and the various variants of the SHA fingerprinting scheme when applied to determine text similarity as seen in [23], similar evaluations were not found for the Rabin's fingerprinting schemes.

2. **Optimal n-gram size**

   After reviewing available information from varied sources, it became apparent that there is no information regarding the n-gram size that would ensure the maximal effectiveness of a plagiarism detection tool that utilizes the n-grams approach in its operations.

3. METHODOLOGY OF THE STUDY

The research data that this study used to reach its conclusions was derived from the operation of a prototype application that the researcher developed. The application was evaluated on its ability to detect instances where some or all of the content in a submitted document was similar to that of documents scanned into its reference index prior. The performance of the application was determined by measuring and recording running and response times for various constituent processes within the application, all measured to microsecond precision.

The prototype application was built as Java EE platform, 3-Tier architecture application comprising of a web application

component, a data store component and a client component. It was built in modular manner such that all the 3 tiers can be deployed on the same computer or they can be deployed on different computers systems without any need for further development.

3.1 Creation of the reference index

The initial step in the creation of the prototype application was the development of the component that creates and maintains the reference index. Documents targeted for plagiarism evaluation are evaluated against the content of the reference index. In its operation, this component examines submitted documents, extracts all the readable content from them, parses them into readable sentences, generates n-grams from them and stores them into an indexed database.

3.1.1 Research concepts

1. Apache Tika
   This is an open source text processing library by the Apache Software Foundation as seen at [1]. It was used in this research for the use cases listed below:

   -Establishing the language a document has been prepared in. The research focused on plagiarism detection in the English language.

   -Determining the file format a document is presented in.

   -Extracting the textual content in the accepted file formats.
2. Alias-I LingPipe
   This is a Natural Language Processing utility [4] that has been used in extracting sentence fragments from readable file contents as derived from Apache Tika detailed above.
3. MongoDB
   This is DBMS system type that stores the data that forms the reference index. MongoDB [5] is a document oriented DBMS that's highly scalable when used to store text based documents and where transactional integrity is not an overriding concern. This was suitable for this research as the reference index largely consists of unchanging text streams of data and there is no urgent requirement for transactional integrity.
4. Middleware
   To manage database CRUD and query interactions, the research employed Hibernate OGM and Hibernate Search.

   -Hibernate OGM [2] is employed in managing the CRUD operations between the various application components and the MongoDB database.

   -Hibernate Search [3] provides enhanced query performance by supplementing the database indexing and retrieval capabilities.

3.1.2 Entity Models

The following entity models were used within the prototype application for defining database table structures and information interchange structures within the components of the prototype application.

1. Reference Document: This represents an instance of a file document digested into the reference index.
2. Document Sentence Fragment: This is an instance of a linguistically complete sentence derived from a reference document.
3. Sentence N-Gram Fragment: This is a single instance of an n-gram partition of the words within a Document Sentence Fragment

3.2 N-Gram and tokenization operations

The research examined the use of plain n-grams and rolled n-grams and their effectiveness as basis for developing a plagiarism detection tool. The n-gram routines were been implemented by the researcher. Tokenization operations such stripping of unnecessary spaces and punctuation were also been done by the researcher.

3.3 Comparing the performance of the Rabin's Fingerprint, MD5 and SHA-1

The research sought to compare the performance in text fingerprinting functionality between an optimal implementation of the Rabin's fingerprint against MD5 and SHA-1, all done on the Java platform.

3.3.1 Determination of an optimal Java Rabin's fingerprint implementation

Four implementations of the Rabin's fingerprinting scheme were considered as listed in Table 1.

For the implementations listed in Table 1, the average fingerprinting time for document corpus of different sizes was recorded and the implementation with the minimized average fingerprinting time selected as the optimal one and designated it as R0.

3.3.2 Comparing fingerprinting time for R0, MD-5 and SHA-1

Comparison was performed for average text fingerprinting time between R0, MD-5 and SHA-1. This comparison was done on n-grammed text content derived from document corpus of various sizes with the N-Gram size used being 8. As will be shown in the results and findings discussion, R0 was established to perform better than MD-5 and SHA-1.

Table 1: Examined Java Rabin's fingerprint implementations

Rabin's Fingerprinting Scheme Assigned Name	Implementation Sourced From	License Type
Bill Dwyer's Implementation	https://github.com/themadcreator/rabinfingerprint	Apache License, Version 2.0
R Stata's Implementation	https://sourceforge.net/projects/rabininjava/	Apache License Version 2.0
S Rowen's Implementation	https://sourceforge.net/projects/rabinhash/	GNU GPL License version 2.0
Yi, Hangehee's Implementation	https://github.com/javarouka/WebCralwer/blob/master/src/prototype/crawler/util/RabinHash.java	Not Available

3.3.3 Determination of process durations

For research activities where calculation of average running times for various processing activities was required, the duration taken for the process was evaluated using the approach detail below

Process Duration (microseconds) =
System Time at Process End - System Time at Process Start (nanoseconds) / 1000

3.4 Determination of the optimal n-gram size

The optimal N-gram size was calculated as the size of n-gram that minimized on the time taken to create n-grams from reference document-corpus of different sizes for plain-grams and rolled N-grams. The index sizes considered were between 1 and the average length in words for sentences in the documents the research was conducted on which was 21.

Total N-gram Time (microseconds) = Average fingerprinting time for plain n-grams + Average fingerprinting time for rolling n-grams.

Selected the most optimal fingerprinting size as the one that minimized on the Total N-gram Time as defined above.

3.5 Determination of the impact of n-gram rolling on plagiarism detection effectiveness.

This was determined in a twofold process that sought to determine whether the use of rolled n-grams added to the amount plagiarized content detected.

1. Using a single randomly selected document, created a reference index using the plain n-grams approach. Using the same document created rolled n-grams and selected 15 n-grams from the result, and then determined how many of them could be recovered from the reference index.

2. Using a single randomly selected document, created a reference index using the rolled n-grams approach. Using the same document created plain n-grams and selected 15 n-grams from the result, and then determined how many on them could be recovered from the reference index.

As discussed in the results and outcomes, the rolled n-grams approach provided more effectiveness in ability to recover any possible n-grams of a given size from a target document.

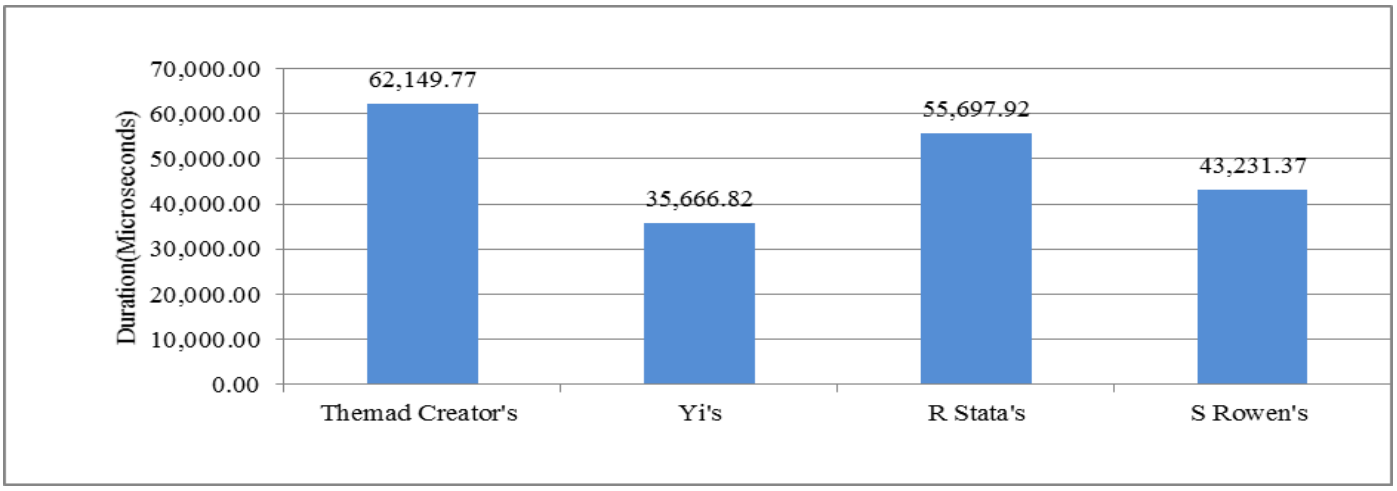
4 FINDINGS AND RESULTS

4.1 Selection of an optimal Rabin's fingerprinting implementation

As detailed previously, four Java based implementations of the Rabin's fingerprint were considered and were subsequently used to derive file level fingerprints on a corpus of 30 randomly selected documents. Fig 1 illustrates the results that were recorded for this activity.

From the graphed results in Fig 1, Yi's implementation was selected as the most optimal Java Rabin's fingerprinting scheme implementation among the four considered as it minimizes on the average fingerprint time. It was designated as R0 and carried forward to the follow-up steps of the research.

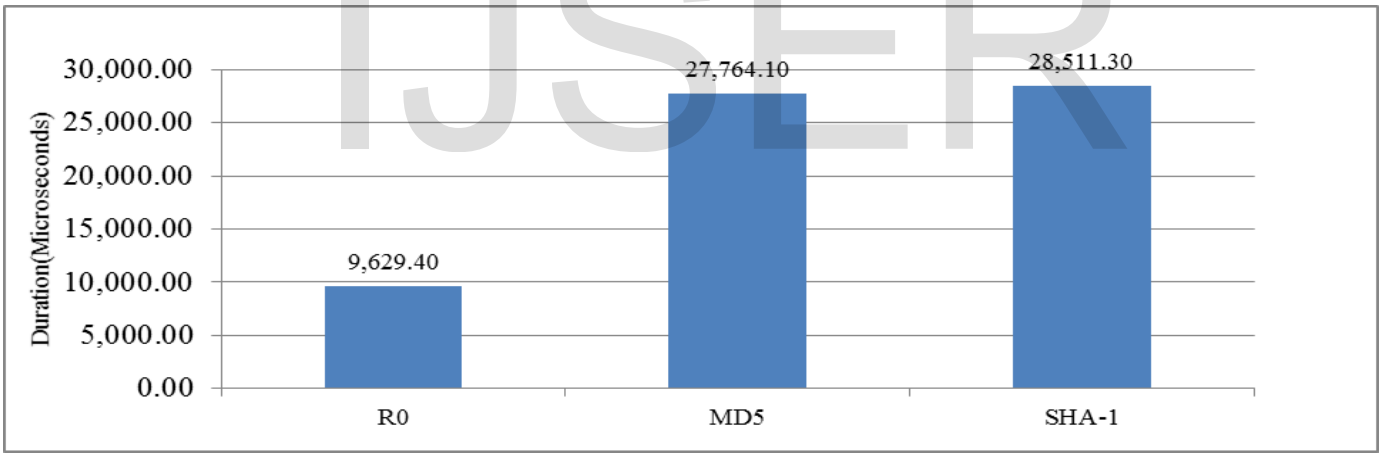
Fig 1. Various Java Rabin's fingerprint implementation average fingerprinting time (Microseconds)



4.2 Comparison of fingerprinting efficiency of R0 versus MD5 and SHA-1

After establishing a suitable Rabin's fingerprint implementation, the research next considered its fingerprinting performance as compared to that of MD5 and SHA-1. The findings for this research activity are presented in Fig 2.

Fig 2. Average fingerprinting time (R0 vs MD5 vs SHA-1) in (Microseconds)



Advantage factor for **R0 vs MD5** = (27,764 / 9629) = 2.88

Advantage factor for **R0 vs SHA-1** = (28,511 / 9629) = 2.96

From the foregoing, **R0** representing the Yi's Rabin fingerprint implementation is observed to yield the minimized average text fingerprinting time in comparison to MD5 and SHA-1 on a document corpus of size 30 and default n-gram size of 8. R0 is observed to outperform MD5 and SHA-1 by factors of 2.88 and 2.96 respectively. Similar results were arrived at for document corpus and n-grams of different sizes. Question 1 of the research is thus answered in the affirmative, for the fingerprinting task as might be used in a plagiarism detection tool, the Rabin's fingerprinting scheme provides better computational performance as compared to MD5 and SHA-1 on the same hardware platform.

4.3 Determination of the optimal n-gram size

The optimal N-gram size was determined as the size of n-grams that minimized on the time taken to create n-grams from a randomly selected reference document for both plain and rolled N-grams. The index sizes considered were between 1 and the average length in words for sentences encountered in the documents the research was conducted on which was 21.The results for this activity are discussed below.

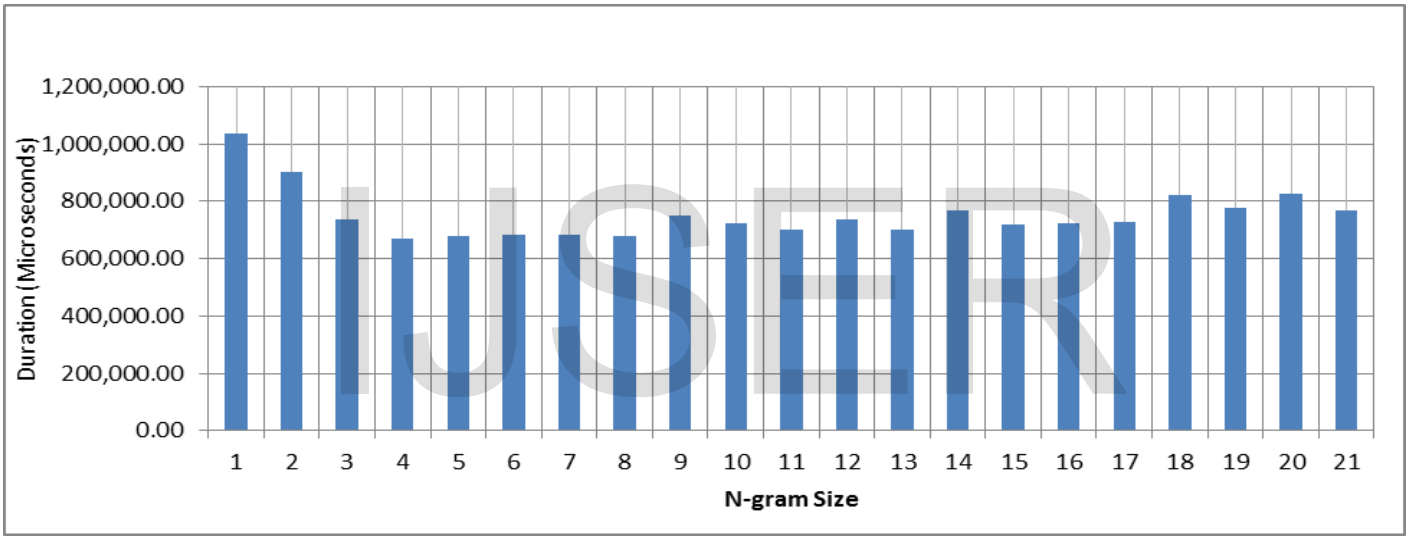
Table 2: Plain + Rolled N-grams generation time for a target document and different n-grams sizes

n-gram size	1	2	2	4	5	6	7
Time (Micro Seconds)	103,558	90,045	73,732	66,7831	67,728	68,337	68,519

8	9	10	11	12	13	14	15	16	17
67,844	74,803	72,217	70,036	73,670	69,924	76,803	71,684	72,229	72,854

18	19	20	21
82,182	77,807	82,675	76,988

Fig 3. Plain + Rolled N-grams generation time for a target document and different n-grams sizes



From the foregoing, an n-gram size of 4 was selected as the most effective.

4.4 Determination on whether the inclusion of rolling n-grams improves on the plagiarism detection effectiveness.

This was accomplished by evaluation of n-gram recovery rates in a twofold manner.

1. Examining the rate of recovery of plain n-grams fingerprints from a reference index comprising of rolled n-grams fingerprints only. For this case, the results established were as detailed in Table 3.

2 Examining the rate of recovery of rolled n-grams fingerprints from a reference index comprising of plain n-grams fingerprints only. For this case, the results established were as listed in Table 4.

Fig 4 provides a graphical comparison of the recoverly rates for the two cases as explained above.

Advantage factor for Rolled n-grams versus Plain n-grams = 12.6/4.16 = 3.02

From the foregoing, it can be observed that a reference index comprising of fingerprints of n-grams generated using the rolling n-gram approach provides a higher recovery by a factor of 3.02 as compared to one comprising of fingerprints of plain n-grams. Thus, n-gram rolling is a useful adaptation that can be added to the n-gramming process to increase the effectiveness of an n-gram based plagiarism detection application.

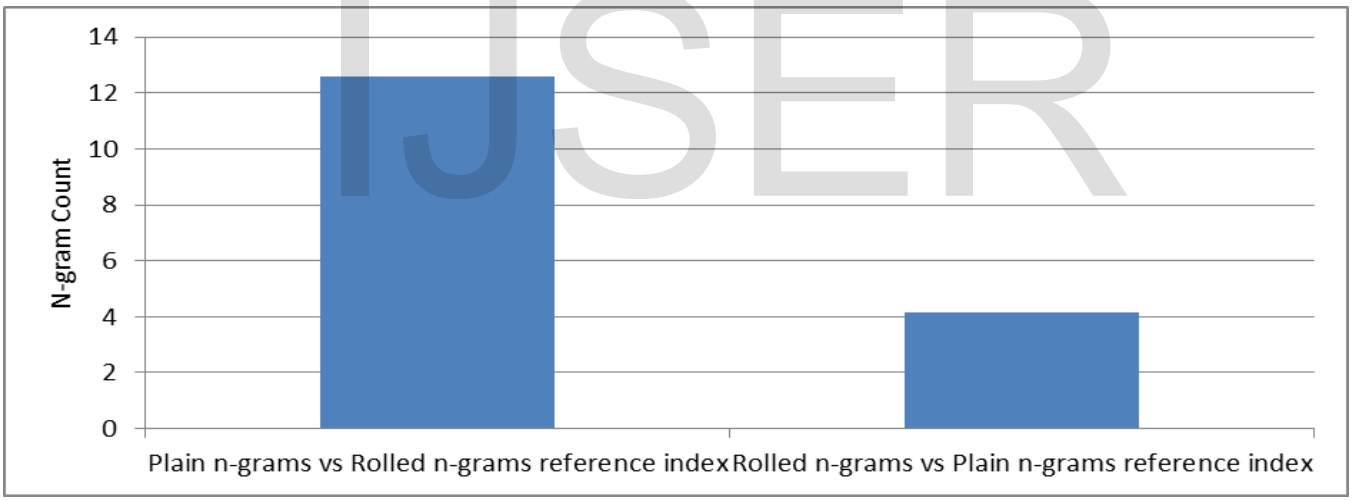
Table 3: Recovery of plain n-grams from a rolled n-grams only reference index

No of recovery runs	30
No of plain n-grams attempted recovery for	15
Average rate of recovery	12.6

Table 4: Recovery of rolled n-grams from a plain n-grams only reference index

No of recovery runs	30
No of rolled n-grams attempted recovery for	15
Average rate of recovery	4.16

Fig 4: Recovery of plain n-grams from a rolled n-grams only reference index vs recovery of rolled-ngrams from a plain n-grams only reference index.



5 CONCLUSION AND RECOMMENDATIONS

This research explored the use of the Rabin's fingerprint and adaptive n-gram techniques as basis for creating a high performance plagiarism detection tool. To facilitate the research, a prototype application was created based on the Java platform and was later used to derive the different kinds of research data required and from which useful conclusions were drawn.

5.1 Conclusions

1. For the plagiarism detection task, Rabin's fingerprinting scheme is the more efficient fingerprinting scheme as compared to MD5 and SHA-1.

2. The n-gram size that was found to maximize the performance and effectiveness of the prototype application was that of 4 words per n-gram.

3. The research established that the addition of n-gram rolling greatly enhanced the ability of the prototype application in detecting plagiarism.

5.2 Recommendations

1. Further research is required to determine the performance and effectiveness of the prototype application where the reference index comprises of a document corpus that potentially includes most of the currently published resources that may be referenced in any academic work, this could span into billions of documents.

2. Further research is required to determine how semantic tools like WordNet can be employed in making plagiarism detection tools more effective by facilitating checks for word and phrase replacement as a way of avoiding plagiarism detection.

REFERENCES

[1] Apache tika. https://tika.apache.org. Accessed: 2016-09-30.

[2] Hibernate ogm. http://hibernate.org/ogm/. Accessed: 2016-09-30.

[3] Hibernate Search. http://hibernate.org/search/. Accessed: 2016-09-30.

[4] Lingpipe. http:// alias-i.com/lingpipe. Accessed: 2016-09-30.

[5]Mongodb. https://www.mongodb.com/. Accessed: 2016-09-30.

[6] Bille, P., Cording, P. H., Gortz, I. L., Sach, B., Vildhoj, H. W., and Vind, S. (2013). *Fingerprints in Compressed Strings*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[7] Black, J., Cochran, M., and Highland, T. (2006). *A Study of the MD5 Attacks:Insights and Improvements*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[8] Brennan, M., Afroz, S., and Greenstadt, R. (2012). Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Trans. Inf. Syst. Secur.*, 15(3):12:1–12:22.

[9] Daelemans, W. (2013). *Explanation in Computational Stylometry*, pages 451–462. Springer Berlin Heidelberg,Berlin, Heidelberg.

[10] *Napolitano V. Princeton Univ. Trustees* (1982).

[11] Gipp, B. (2014). *Citation-based Plagiarism Detection: Detecting Disguised and Cross-language Plagiarism Using Citation Pattern Analysis*. Springer Vieweg.

[12] Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. EBL-Schweitzer. Cambridge University Press.

[13] Guthrie, D., Allison, B., Liu, W., Guthrie, L., and Wilks, Y. (2006). A closer look at skip-gram modelling. In *In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006*, pages 1222–1225.

[14] Hiremath, S. and Otari, M. Plagiarism detection - different methods and their analysis : Review. *International Journal of Innovative Research in Advanced Engineering*, 1(7).

[15] Hoad, T. C. and Zobel, J. (2003). Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.*, 54(3):203–215.

[16] Juola, P. (2006). Authorship attribution. *Found. Trends Inf. Retr.*, 1(3):233–334.

[17] Karp, R. M. and Rabin, M. O. (1987). Efficient randomized pattern-matching algorithms. *IBM J. Res. Dev.*,31(2):249–260.

[18] Michailidis, P. D. and Margaritis, K. G. (2001). String matching problem on a cluster of personal computers: Experimental results. In *in Proc. of the 15th International Conference Systems for Automation of Engineering and Research*, pages 71–75.

[19] Mnookin, S. (2010). Setting the agenda: The New York Times jayson Blair report and its impact 45 on American media.

[20] Muthitacharoen, A., Chen, B., and Mazi`eres, D. (2001). A low-bandwidth network file system. In

Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP '01, pages 174–187, New York, NY, USA. ACM.

[21] Rabin, M. (1981). *Fingerprinting by Random Polynomials*. Center for Research in Computing Technology: Center for Research in Computing Technology. Center for Research in Computing Techn., Aiken Computation Laboratory, Univ.

[22] Salton,G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

[23] Stein, B. and Meyer, Z. E. S. (2006). *Near Similarity Search and Plagiarism Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[24] Sursock, A. (2015). *Trends 2015:Learning and Teaching in European Universities*.

[25] Tiersma, P. and Solan, L. (2012). *The Oxford Handbook of Language and Law*. Oxford Handbooks. OUP Oxford.

[26]Tirilly, P., Claveau, V., and Gros, P. (2008). Language modeling for bag-of-visual words image categorization. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, CIVR '08, pages 249–258, New York, NY, USA. ACM.

[27] Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188.

[28] Wang, K., Thrasher, C., Viegas, E., Li, X., and Hsu, B.-j. P. (2010). An overview of Microsoft web n-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, HLT-DEMO '10, pages 45–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

[29] Wang,X., Yin, Y. L., and Yu, H. (2005). Finding collisions in the full sha-1. In *Proceedings of the 25th Annual International Conference on Advances in Cryptology*, CRYPTO'05, pages 17–36, Berlin, Heidelberg. Springer-Verlag.