Document Fingerprint Detection System


Department of Information Technology, Sri Lanka Institute of Information Technology, Malabe

DFDS is developed to prevent copying someone else’s work as your own work which is known as “Plagiarism”. Plagiarism is a growing problem in all universities, colleges and schools today. Detection can be done either manually or in a computer-assisted manner. Manual detection requires substantial effort and excellent memory to identify, and it is impractical in cases where too many documents must be compared, or original documents are not available for comparison. Computer-assisted detection allows vast collections of documents to be compared to each other, attempts to obtain, make successful detection. By using DFDS one is able to find copying that the student has done in assignments from others and online web documents. To improve producing of most accurate results than existing systems, Document Fingerprint Detection system has used most suitable hash function, algorithm, searching method and comparison methods. System takes student assignments as inputs. Sends the document to the parsing module. It clears unnecessary punctuation marks, white spaces, letter cases and refines the document. Token manager takes input as a refined document and then the document is divided into Tokens. Creates hash values for each token and generates fingerprint which selects some tokens out of all tokens. Those tokens are sent to web search engine. The system searches suspect documents that is similar to assignment. System downloads some suspect documents from web and compares local student documents with those downloaded documents. Finally generates percentage of own work, web copied work, suspect work, referred text, percentage of friend’s work and student identity numbers to detect copied from original document.

Index terms - DFDS- Document Finger Print Detection System, k-gram, Tokens, Token step, Winnowing Algorithm

I. INTRODUCTION

Plagiarism is the copying of one’s work or ideas into your own work without full acknowledgement. There are different types and different degrees of plagiarism which are; submission of work not written and prepared by students. The system searches suspect documents that are similar to the assignment. Downloads some suspect documents from the Web and compares the local student document with those downloaded documents. Academic staff can monitor answer papers and check correctly whether documents are copied or not. It’s fair to other students who have actually answered without plagiarizing. The main objective of the DFDS is to encourage students to engage good practices and morals in writing answers by prevent copying from others.

A. Background

Copying or stealing the work of another student’s doing directly from a text word-for-word, or to use an attractive phrase or sentence they have found somewhere. Purchase existing written work, use text downloaded from the Internet not only text also such as pictures, photographs, or diagrams without acknowledging sources. Before introducing the Document Finger Print Detection System, it is a part to discuss little bit about existing Plagiarism Systems. Existing Plagiarism Systems, WCopyfind 2.5v is a simple standalone software that can detect local plagiarism. It’s able to detect if a certain file is partially or fully copied from the 2nd file or both files partially or fully copied from the 3rd document. It cannot search Internet documents. The Project after 2003 wasn’t updated. Plagiarism Detector is a strong plagiarism detection application that detects web and local plagiarism. Good end user interface and easy to handle. Gives well understandable reports. High cost to buy. SPLAT system designed to detect self plagiarism. Uses Webs spider to crawl through the websites of the top fifty computer science departments and downloading research papers and checks textual reusing by text comparison algorithms. But DFDS mainly addresses to detect plagiarism on both online searching and local textual scanning. Document Comparison Module is efficient. Using the system can identify dependencies among documents, can highlight plagiarized portions and can get average of copied works. DFDS is easy to maintain, install and it is user friendly.

II. ARCHITECTURE

Document Fingerprint Detection System has three modules. They are:

1. Document Parsing Module
2. Web document Parsing Module
3. Document Comparison Module

A. Document Parsing Module

Document parsing module has following sub modules. They are structural characteristic generation, divide document in to k-grams and generates hashes for each and every k-gram. The
system removes three desirable properties which effects detect its own identification. Namely they are whitespace Insensitivity, noise suppression and positional independence. In matching text files, matches should be unaffected by such things as extra whitespace, capitalization, punctuation, etc. Here whitespace and punctuation are removed and all letters are converted in to lower cases. Noise suppression means there are many common words like the, is, are etc. and idioms. This is also removed and continuous sentences will be sent in to k-gram module, where k-value will be chosen by the user. System automatically continues to divide sentences in to substrings.

As an example let’s assume that the string which it’s passing into the process is “adorunrunrunadorunrunrunrunru runrunru”. The variable k that what we’ve defined in the above (k-gram) can assign any value by the user. So, let’s take the value as k=5, so then the algorithm will divide the document into substring of length 5 and the final result will be shown like this.

“adoru dorun orunr runru unrun nrnun runad unado nador adoru dorun orunru runru unrun”

After dividing document in to k-grams it will be sent immediately to hashing module. There it will generate the unique hash values for every k-gram. It will give the same hash value for the same k-grams and the values are identical.

“adoru dorun orunr runru unrun nrnun runad unado nador adoru dorun orunru runru unrun

77 72 42 17 98 50 17 98

nrnun runad unado nador adoru

8 88 67 39 77
dorun orunru runru unrun”

72 42 17 98

From the hashed k-grams it selects some subset of these hashes to be the document’s fingerprints. A fingerprint also contains positional information, which we do not show, describing the document and the location within that document that the fingerprint came from. If the hash function is chosen, the probability of collisions is very small, and then whenever two documents share one or more fingerprints, it is extremely likely that they share a k-gram as well. Need an efficient algorithm for selecting the fingerprints from a sequence of hashes that guarantees that at least part of any sufficiently long match is detected. By considering that, use the **Winnowing algorithm** for the purpose of generating fingerprints from hash of k-grams. [1]

### B. Web Document Parsing Module

Take filtered document from structural Characteristic Generator. Pass filtered document to Token Generator. What Token Generator is doing is to extract tokens to Google search. (Token is N length amounts of words from given set of words.) There the system will miss predefined number of tokens. It is called Token Step. From Google search engine it returns set of suspected URLs. Download top 5 web links. Before system download URL the system checks whether system already download given URLs in same project by doing this can eliminate duplicate downloads. Downloads web pages for all tokens then filters downloaded web pages. As usual web pages have advertisements, graphics etc that are not needed. Need to extract wanted text from web pages. Because of html tags system can’t read the web page as normal text file. After filtered, send web documents to make web fingerprints and save in database till compared with local documents.

### C. Document Comparison Module

DFDS has steps as comparison. System compares suspect documents with a large collection of other documents and attempts to match similarities. It means that catches these cheats matching textual data with particular websites and documents which are locally submitted. This tool makes comparison among documents and delivers a report mentioning percentage of plagiarized work, suspect work, own work and referred text percentage and percentage of friends work and friend’s student’s identity numbers. The system can detect who has copied from the original document.

### III. RESEARCH FINDINGS

The whole development of DFDS isn’t a full research. But there are some various research parts. Under this section all those research achievements, are described.

1. Use of winnowing algorithm
   Document Fingerprinting Detection system uses Winnowing algorithm to select fingerprints from hashed of k-grams.

2. Efficiency of the system
   DFDS does not require to register documents and it removes all documents after being checked over. E.g. assume A, B, C documents for check and A document compares with B, C documents. If A document has not been detected as plagiarism document, it strips from the system.
3. Time consumption
Existing systems take more time to detect plagiarism due to inefficiency of their algorithm. Use winnowing algorithm and it does not take more time to compare documents.

4. Defining k-values, Token length and Token step
In the system user can define those values. In the Document Parsing Module k-value is defining. In Web Document Parsing Module define Token length and Token Step. These values depend on the accuracy, efficiency and time consumption.

IV. Evaluation
Implemented all solutions in Microsoft Visual Studio 2008 C# and experiments are performed on a PC with DMI Wistron 3617 + dual core CPU (2.00 GHz) and 2GB RAM. The operating system is Windows Vista.

This illustrates how it will affect the K_value for the final outcome within the Document Parsing Module.

Document Size - 46.7 KB
Total Number of Letters – 19263

![Fig.2. - K_value variation with time](image)

According to the above depiction can analyze, depending on the k-value the time taken for the final outcome is different. Here the main objective is to find out the optimized k-value range which will be most effective and suitable for the comparison level.

For testing use a normal text document (size of 46.7KB), Time measured in seconds and used k-value range is (1 - 15).

Also during the research the team found a new relationship between k-value, k-grams and total number of letters in filtered document.

\[
\text{Total Number of k-grams} = \text{Total Number of Letters} - \text{k-value} + 1
\]

Depending on the K_value it determines the final outcome. Take the optimal solution here it’s very important to find out the most suitable value as the K_value.

<table>
<thead>
<tr>
<th>Document Name</th>
<th>File size (KB)</th>
<th>TNL</th>
<th>TNW</th>
<th>TNL/TNW</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>46.7</td>
<td>1926</td>
<td>1838</td>
<td>5.019030123</td>
</tr>
<tr>
<td>B</td>
<td>7.58</td>
<td>1118</td>
<td>620</td>
<td>5.691290323</td>
</tr>
<tr>
<td>C</td>
<td>6.51</td>
<td>2715</td>
<td>521</td>
<td>5.255599898</td>
</tr>
<tr>
<td>D</td>
<td>5.02</td>
<td>2232</td>
<td>456</td>
<td>4.605594516</td>
</tr>
<tr>
<td>E</td>
<td>1.1</td>
<td>1304</td>
<td>270</td>
<td>5.031531852</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>25.0463589</strong></td>
<td><strong>Average</strong></td>
<td><strong>5.0913798</strong></td>
</tr>
</tbody>
</table>

According to the above diagram, chosen 5 text documents and among those documents can find the most suitable K_value.

First need to calculate TNL, TNW, TNL/TNW values for each document and the average value of the \( \sum (\text{TNL/TNW}) \). Finally by substituting those values into following equation can find out the optimal K_value.

\[
K_{\text{value}} = \left[ \frac{\sum_{n=1}^{n} (\text{TNL/TNW})}{n} \right] - 1
\]

n – Total number of documents

Let’s consider the following example,

<table>
<thead>
<tr>
<th>Name</th>
<th>File size (KB)</th>
<th>K_value</th>
<th>Time[s]</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>test 1</td>
<td>69.71</td>
<td>3</td>
<td>31.79584159</td>
<td>5</td>
</tr>
<tr>
<td>test 2</td>
<td>69.71</td>
<td>4</td>
<td>46.1484711</td>
<td>4</td>
</tr>
<tr>
<td>test 3</td>
<td>69.71</td>
<td>5</td>
<td>54.6390319</td>
<td>2</td>
</tr>
<tr>
<td>test 4</td>
<td>69.71</td>
<td>6</td>
<td>55.7479715</td>
<td>2</td>
</tr>
<tr>
<td>test 5</td>
<td>69.71</td>
<td>7</td>
<td>63.6029757</td>
<td>2</td>
</tr>
<tr>
<td>test 6</td>
<td>69.71</td>
<td>8</td>
<td>67.39667912</td>
<td>2</td>
</tr>
<tr>
<td>test 7</td>
<td>69.71</td>
<td>9</td>
<td>69.6518189</td>
<td>2</td>
</tr>
<tr>
<td>test 8</td>
<td>69.71</td>
<td>10</td>
<td>72.0816809</td>
<td>2</td>
</tr>
</tbody>
</table>

The above diagram illustrates some test cases and relevant output for each scenario. By applying this value to the above equation found that the K_value = 4, that means the optimal solution is given by the test 2.

In addition considering the following graph can define the optimal solution given when the K_value = 4 and the optimal time taken for the comparison is equal to 46.1484711 seconds.
V. CONCLUSION AND FUTURE WORK

DFDS is implementing to give an opportunity and the facility to the user to use in a convenient manner. As further researches it is carrying out to generate most accurate results to user to determine exact plagiarized documents and give percentage of plagiarized work. With use of impartial computer system and students who did plagiarize tend not to dispute the results of the plagiarism detector if confronted with the evidence. The success of the project can be measured in the number of advancements made during the development as well as in the research part.

The presented system can be improved in the following ways.

1. System is used to detect text based assignments. In further developments can develop the system to detect plagiarized images.
2. Improve the system to identify synonyms and writing patterns when comparing submitted documents.

REFERENCES