An Online System for Plagiarism Detection

Vaughn M. Segers, James Connan
Department of Computer Science
University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa
Telephone: +(27) 21 959-3010, Fax: +(27) 21 959-3006
Email: 2435465@uwc.ac.za, jconnan@uwc.ac.za

***Abstract*—This paper discusses the creation of an online system for plagiarism detection. This is a plagiarism detection system which searches the Internet for evidence of plagiarism within a document. This is achieved using the PHP scripting language in conjunction with the Google Internet search engine and various Linux applications. Two methods contained within the system are explained and their varied methodologies outlined.**

***Index Terms*—Plagiarism Detection, Google, regular expressions**

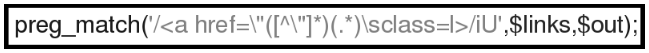
I. Introduction

PLAGIARISM plagues institutions of learning such as schools and universities [7] [4] as access to digital media via the Internet increases [5]. Learners now have the tools and know-how to copy documents from online sources. It is for this reason that a simple and effective method is needed to combat this growing problem. At present educators use cumbersome Internet search engine methods to check for plagiarism. This method involves manually analyzing the document and relying on previous experience to choose the sentences to check with an Internet search engine. A system to automate this would greatly improve the effectiveness and impact of such analysis.

II. Present Approaches

At present, many approaches exist to combat plagiarism. These include applications such as TurnItIn.com [11], the SID [13] plagiarism detection website, Ferret [2] and the Sherlock [8] plagiarism detector to name a few. Other plagiarism detectors also exist with a more specific purpose, such as to determine plagiarism in computer source code. Examples of these are MOSS [14], which supports a wide variety of programming languages, and SID [13], which only supports Java and C++.

Many of these plagiarism detection methods give an outline or even a full explanation of how their plagiarism detection services work. Unfortunately, the most well known and widely used of these systems [3] [4], namely TurnItIn.com, does not elaborate on their method of plagiarism detection. It is known that TunItIn.com determines plagiarism by searching for matches on the Internet and in their own saved document archives [11]. However the comparison methods used are unknown. SID uses Software Integrity Detection technology which computes the level of shared information between two documents. The Ferret plagiarism detector uses a method that involves the comparison of trigrams to determine the authenticity of the document. Sherlock turns the text into digital signatures.



```
preg_match('/<a href=\"([^\"]*)(.*)\sclass=l>/iU',$links,$out);
```

Fig. 1. PHP code snippet for the extraction of a link from a Google^{TM}results page

The copy detection systems used to analyze source code for plagiarism [14] [13] generally work differently to those which detect plagiarism in ordinary text documents. Code comparison applications [14] [8] make use of tokenization [12] of variables as part of the detection process.

III. Pre-Development

Careful consideration was given to the previously mentioned methods of plagiarsm detection and the following design decisions made. The system is a fully web-based sever-side program. The LAMP (Linux Apache MySQL PHP) approach was used for the system. Using LAMP is beneficial as the LAMP components are free and open source.

PHP is a sever-side web scripting language which produces HTML for the client browser. PHP has the benefit of placing the workload on the server and not the client computer, as with other scripting languages such as JavaScript. For this plagiarism detector the ability of PHP to handle regular expressions is also useful as seen in Figure 1.

GNU/Linux applications that were used are *w3m* and *ps2ascii*. W3m is a text based web browser. All interfacing with the Internet is done by the use of w3m. Ps2ascii is a GNU/Linux program which converts PDF and PostScript (PS) files to ordinary text. Document conversion using ps2ascii for the uploaded documents is necessary for the system. MySQL was used to perform the database functions in conjunction with PHP. MySQL is a database management system.

Google was the online search engine used due to the wide use shown in [10] against other search engines. Google also provides options to change the parameters of a search within

J. Connan is with the Department of Computer Science, University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa (phone: +(27) 21 959-3010; fax: +(27) 21 959-3006 ; e-mail: jconnan@uwc.ac.za).

V. M. Segers is a M.Sc student in the Department of Computer Science, University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa (e-mail: 2435465@uwc.ac.za).

the search query, such as specifying the filetype returned. This assists in the necessary search refinement for this plagiarism detector. This project is completely Internet-dependant as the only source of comparison documents. As such, larger documents are bandwidth-intensive and this load is placed on the server only.

The system incorporates these programs the following manner:

1) File upload and user interaction is managed using PHP.
2) Ps2ascii is used to convert uploaded files.
3) Uploaded files are stored in the MySQL database.
4) PHP uses information in the MySQL database to form Google queries.
5) W3m is used for interaction with the search engine and retrieval of query results.
6) PHP regular expressions are used to filter query results.
7) PHP is used to enter web results into the MySQL database and collect the plagiarism detection information.
8) The database information and plagiarism results are then displayed in HTML to the user by PHP.

IV. DEVELOPMENT

The system was created to be as simple as possible for the user to interact with. Two types of plagiarism detection are offered that provide different levels of search refinement, Robust Detection and LightWeight Detection. The differences between the two will be outlined below. The interface was left plain to focus on the core functionality.The system flow is shown in Figure 2 .

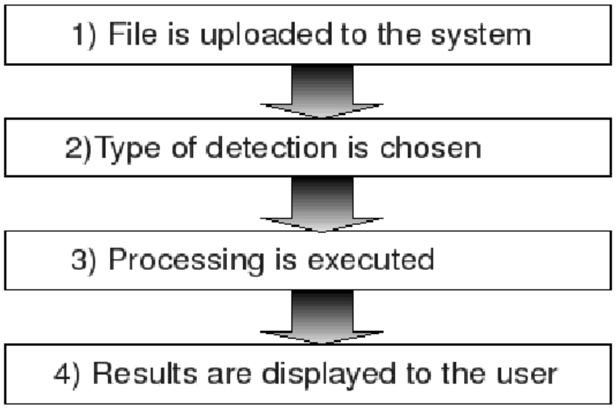

1) File is uploaded to the system
2)Type of detection is chosen
3) Processing is executed
4) Results are displayed to the user

Fig. 2. Basic steps to use the plagiarism detection system

A. Robust Detection

The purpose of Robust detection is to check documents for plagiarism using the Internet. It can be said with a high degree of certainty that online matches found using this method have probably been plagiarized.

1) After the file has been uploaded it is checked for malicious data by analyzing the file size.
2) The following task is to remove all punctuation in the document and to convert the text to lowercase for document consistency. This is done by using regular expressions generated with PHP.
3) The next step is based on the research done by Knight, Almeroth and Bimber [6]. The text is searched for all sentences which are greater than eight words in length. The number eight was chosen as this produced the least false-positives [6] in their research.
4) Google is then used in the search for these sentences. This is done by generating Google queries with these specific sentences within inverted commas, thus the entire sentence is searched for. These results are then filtered using the regular expression functionality of PHP to extract the links from the Google results page.
5) The results will then be displayed to the user with links to the online documents found.

B. LightWeight Detection

This method of detection will also compare content to online documents, but seeks to employ more indirect methods in the pursuit of more subtle occurrences of plagiarism.

Trigram comparison [9] [1] is performed in LightWeight detection. Trigrams measure similarity between two documents. Documents for comparison are found as follows:

1) After the file has been uploaded it is checked for malicious data.
2) The document is then scanned to remove all punctuation and the text is converted to lowercase.
3) To perform a more thorough search of online material available, it is now considered that phrases in different places in the text could be plagiarized. Thus a way is needed to determine which groups of words these are. For this purpose it was devised that words which are unique to the document, i.e. which appear only once, are to be sought. These words are then combined with three adjacent words to create a phrase for use in the online search.
4) To further expand the pool of relevant documents a string of three words was found. These three words were each to be greater than five characters in length. Such a triplet in a document was found to be a useful discriminating factor. A search was done to collect documents with these consecutive words.
5) The pool of documents now downloaded are compared to the original user document by the method of trigram searching [9] [1]. For this each document is converted into a set of overlapping three word groups, as represented in Figure 3. Plagiarism is likely if the ratio of matching trigrams is greater than 25 percent between the two texts.
6) The results are then shown to the user with the suspicious trigrams highlighted and linked to the suspected documents.

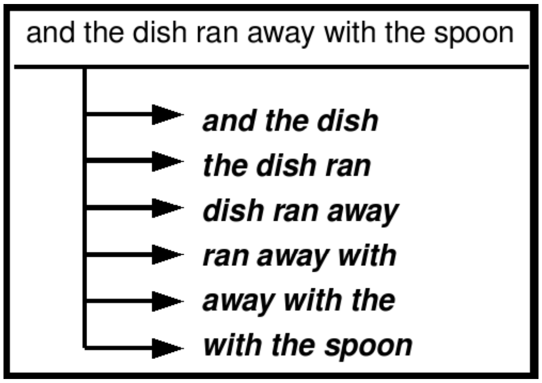


Fig. 3. A sentence split into trigrams.

V. Test Results

Tests were conducted in he same manner on both methods.

A. Plagiarized Test Documents

The test documents were random texts taken from scientific papers found on the Internet. These texts had an average of 308 sentences and comprised of approximately 1151 words each. The plagiarism detector currently only supports the input of text documents and as such all documents taken from the Internet had to be converted to text. To check accuracy against ps2ascii for PDF and PS documents, Google's document conversion facility was used when downloading original texts during testing. This conversion was not accurate and all texts were manually checked for garbage. References were also removed. An overview of the test data used in Robust detection can be seen in Table I.

1) Results on Robust Detection: In all cases the first link found by the system correctly pointed to the document of origin.

Average number of words	1151
Average number of sentences	308
Average number of sentences greater than 8 words in length	18.4
Average time taken to check (seconds)	95.9

TABLE I
Test Data for Robust Detection

2) Results on LightWeight Detection: LightWeight Detection also correctly pointed to the document of origin.

As stated, both methods performed equally well in finding the original document. This result was as expected.

B. Plagiarism-free Test Documents

Documents containing no plagiarism were generated by taking a list of commonly used words and randomly forming a document of similar length to those of the plagiarized test documents.

1) Results on Robust Detection: No documents were found.

2) Results on LightWeight Detection: No documents were found.

Both methods here correctly found no collusion. This was expected of Robust Detection as the possibility of eight random words in a sentence being exactly copied is viewed as extremely rare. LightWeight Detection proved the effectiveness of trigram comparisons by achieving the same result.

C. Documents with minimal plagiarism

Documents with minimal plagiarism were generated by placing a paragraph of plagiarized text into the randomly generated documents.

1) Results on Robust Detection: In all cases the first result linked to the correct text.

2) Results on LightWeight Detection: The documents were found to have no plagiarism by this method

Robust detection is accurate if at least one plagiarized sentence is found. LightWeight detection however is below the 25 percent similarity barrier due to the large amount of random words. It is for this reason that no plagiarism is detected.

D. Plagiarized Test Documents with minor changes

Minor changes were made to words and spelling of plagiarized test documents for the purpose of this test.

1) Results on Robust Detection: Accuracy was reduced to 80 percent when plagiarized documents were modfied in this manner.

2) Results on LightWeight Detection: The first result returned linked correctly to the plagiarized text.

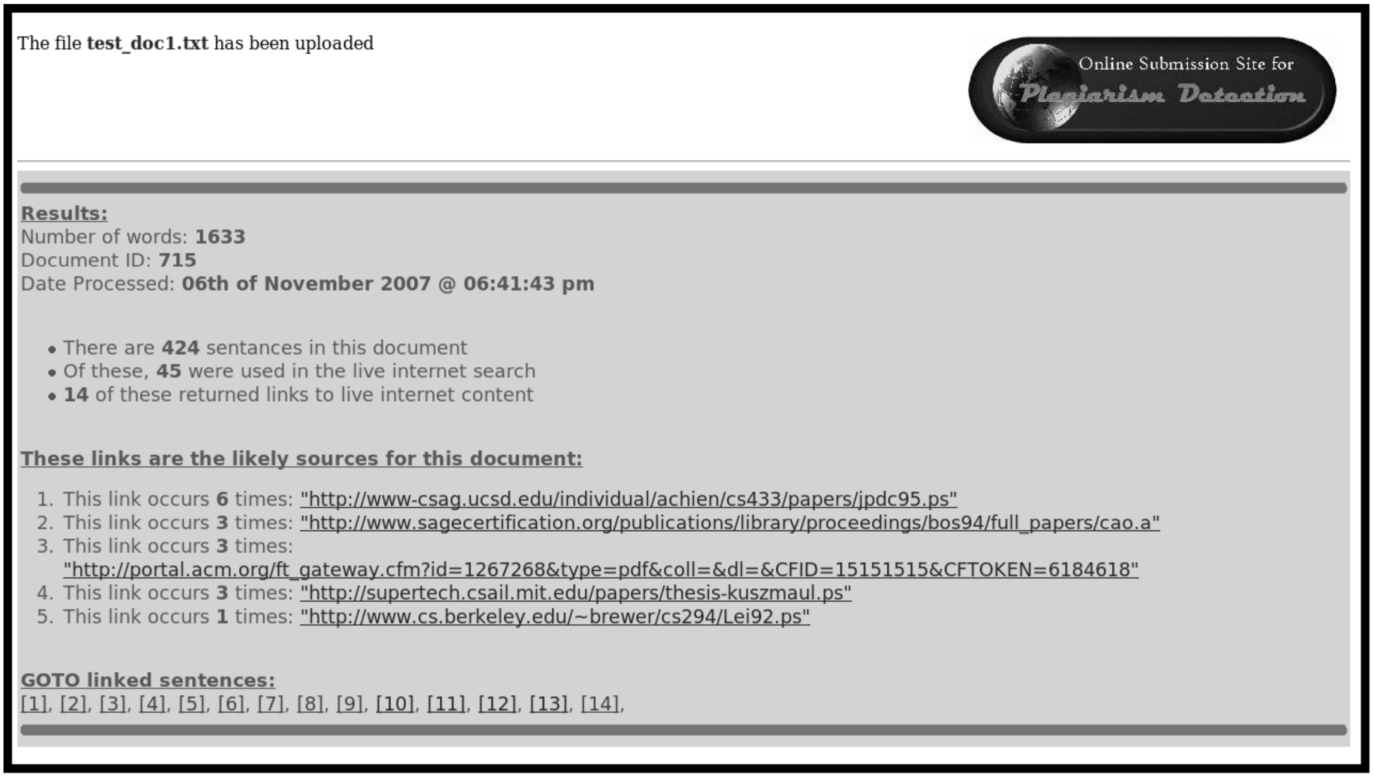
Sufficient plagiarized text is available in both instances to regularly detect plagiarism. Robust detection fails when all sentances greater than 8 words in length have been modified.

It must also be noted that Robust Detection relies solely on Google results for comparisons. LightWeight detection, however, depends on documents to be downloaded before comparison can take place. This directly affects the amount of time taken for detection. Thus the time taken for Robust Detection (on average 95 seconds) is much quicker than LightWeight detection, which can take up to 20 minutes.

A sample of results determined by Robust detection can be seen in Figure 4.

VI. Conclusion

The underlying methods behind these detection systems such as trigram matching provide an effective means to detect plagiarism. Also, the choice of Google as the Internet search tool has proven a great help in this regard as it is their search results that yield such high levels of accuracy. As such this combination makes for effective and easy-to-use plagiarism detection.



The file **test_doc1.txt** has been uploaded

Results:
Number of words: **1633**
Document ID: **715**
Date Processed: **06th of November 2007 @ 06:41:43 pm**

- There are **424** sentances in this document
- Of these, **45** were used in the live internet search
- **14** of these returned links to live internet content

These links are the likely sources for this document:

1. This link occurs **6** times: "http://www-csag.ucsd.edu/individual/achien/cs433/papers/jpdc95.ps"
2. This link occurs **3** times: "http://www.sagecertification.org/publications/library/proceedings/bos94/full_papers/cao.a"
3. This link occurs **3** times: "http://portal.acm.org/ft_gateway.cfm?id=1267268&type=pdf&coll=&dl=&CFID=15151515&CFTOKEN=6184618"
4. This link occurs **3** times: "http://supertech.csail.mit.edu/papers/thesis-kuszmaul.ps"
5. This link occurs **1** times: "http://www.cs.berkeley.edu/~brewer/cs294/Lei92.ps"

GOTO linked sentences:
[1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14],

Fig. 4. A sample of results as determined by Robust Detection

VII. Discussion

LightWeight Detection can be bandwidth intensive and requires a large amount of time for larger documents on slow Internet connections. This is a matter which can potentially be resolved if documents can be filtered before being downloaded. It is envisioned that more new and novel ideas to further prune the process of document download would go a long way to improving the performance of this system.

References

[1] Lyon C. and Malcolm J. and Dickerson B. 2001. Detecting short passages of similar text in large document collections. Proceeds of Conference on Empirical Methods in Natural Language Processing.

[2] Lyon D., Barrett R. and Malcolm J. 2004 . A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector. Plagiarism: Prevention, Practice and Policies Conference.

[3] Letcher M. 2005. Academic Honesty Through Technology.ASCUE Conference.

[4] Burke M. 2004. Deterring Plagiarism: A New Role for Librarians. Library Philosophy and Practice Volume 6, No. 2.

[5] Todd, Benefield, Berens, Borodkin et al. 2005. AN ACT CONCERNING INCLUSION OF AN INTERNET SAFETY PLAN IN EACH SCHOOL DISTRICT'S SAFE SCHOOL PLAN. HOUSE BILL 05-1036.

[6] Knight A., Almeroth K. and Bimber B. 2004 . An Automated System for Plagiarism Detection Using the Internet. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications. pp 3619–3625.

[7] Petress K. 2003 . Academic Dishonesty: A Plague On Our Profession. The Reference Shelf: Intellectual Property. pp 47-50.

[8] Pike R. 2007 .The Sherlock Plagiarism Detector . Retrieved on March 20, 2007, from the University of Sydney website http://www.cs.usyd.edu.au/ scilect/sherlock.

[9] Roberts A. and Morrison A . 2002 .LAPD Language Analysis for Plagiarism Detection. Retrieved on March 20, 2007, from http://www.andyroberts.net/software/.

[10] Nielsen-NetRatings. 2007 .NIELSEN-NETRATINGS ANNOUNCES FEBRUARY U.S. SEARCH SHARE RANKINGS.

[11] iParadigms LLC. 2007. Turnitin. Retrieved on March 20, 2007, from http://turnitin.com/static/home.html.

[12] Mozgovoy M. and Fredriksson K. and White D. and Joy M. and Sutinen E. 2005. Fast Plagiarism Detection System. Proceedings of the 12th International Symposium on String Processing and Information Retrieval. pp 268–271.

[13] Chen X. and Francia B. and Li M. and Mckinnon B. and Seker A. 2004 .Shared Information and Program Plagiarism Detection. IEEE Transactions on Information Theory.

[14] Schleimer S., Wilkerson D. S. and Aiken A. 2003 .Winnowing: local algorithms for document fingerprinting. International Conference on Management of Data.

Vaughn Segers is currently a Telkom Centre of Excellence M.Sc student at the University of the Western Cape. His area of research is focused on applications for the Deaf and hard of hearing.

James Connan heads up the South African Sign Language (SASL) research group. He has a wide range of interests that include: databases, computer vision and machine learning.