

Plagiarex for Digesting Website Text

David Kellogg

December 6, 2004

Abstract

Web archivists need a method to detect significant changes in a document to reduce redundancy in the archive and to document interesting changes. The plagiarex MD5 is created by calculating the MD5 digest of a list of the five longest lower-case words in a document. Examples of plagiarex are given, and suggestions for its use are listed. This technique is extended to Simplified Chinese (GB2312) to show its flexibility with non-Latin character sets. The plagiarex MD5 is used to show significant changes to texts of Shakespeare, Conrad and a Chinese news web site. The algorithm listed here is version *0.6*.

1 Introduction

The problems of tracking web text changes are many. Taking the MD5 of a document is a poor way to track changes in web pages. Other useful techniques exist in detecting changes and plagiarism,¹ but interpretation of text changes can be difficult. MD5 is designed to be very sensitive to changes, even to whitespace and markup tags.

Changes to the date on a page are common. These changes might be quite meaningless to a person viewing the page. Changes resulting from query results confuse spiders, often creating infinite loops. In one case 16,000 local directories were created by a looping spider.

The plagiarex MD5 would stop this. The straight MD5 showed a *different* page for each iteration of the loop, despite the obvious similarities. A function is necessary to track *significant* changes in web pages. The plagiarex MD5 accomplishes this.

¹<http://homepages.feis.herts.ac.uk/~comrcml/plagiarism.01.ps>

2 Constructing the Plagiarex MD5

The steps of creating the plagiarex of a document are listed.

1. Tokenize all words in a document that are not part of markup or computer code.
2. Remove punctuation, such as commas, colons, periods, and others.
3. Reject all words that contain capital letters or any character other than *a* to *z*.
4. List the 5 longest words, starting from the longest, then sorted by original word order, delimited by commas, without spaces.
5. Digest this list using the base64 MD5 algorithm.

3 Distinguishing Revisions in Literature

Four versions of William Shakespeare’s *Hamlet* and two versions of Joseph Conrad’s *Heart of Darkness* are tested for plagiarex signature. This method can detect quickly the large editorial changes that occurred to *Hamlet*.

3.1 Shakespeare’s *Hamlet*

William Shakespeare’s *Hamlet* exists in several versions, including the *First Quarto* and *First Folio*. The goal here is to group and differentiate revisions based on uncommon words. Common words include “the”, “a” and “and.” Uncommon words usually are longer.² Their presence alone can show differences in a work. The *First Quarto* of *Hamlet* begins with Barnardo’s “Stand: who is that?” The five longest words of the *First Quarto* from the University of Victoria’s digitization project³ read “distemperancie”, “entertainment”, “distemperature”, “circumstances” and “indifferently” before digesting. After using an MD5 on the list with commas, but without spaces, this becomes *t968ghH8IC3Yv3n8JYMSVQ*. This is a unique hash of the five words.

²Words over 2 letters long occur less frequently, according to <http://www.blackwell-synergy.com/links/doi/10.1111/j.0039-3193.2004.00109.x/pdf>

³http://ise.uvic.ca/Annex/DraftTxt/Ham/Ham_Q1

It is interesting to compare an independent transcription of the *First Quarto*. From the University of Virginia’s transcription,⁴ the value remains the same, *t968ghH8IC3Yv3n8JYMSVQ*. In both cases, excluded words are “Celestiallbed” and “Guildensterne” due to capitalization. This is a useful exclusion, since Guildensterne is mentioned many times in the text and could have been used in a separate play, had it not been for his untimely death.

What occurs for the *First Folio*, which begins, “Who’s there?” The five words are “encompassement”, “transformation”, “instrumentall”, “stubbornnesse” and “vnderstanding”, which yields a plagiarex MD5 of *EibO4wGSF-rYV+bJC2vxAHA* for both the University of Virginia⁵ and the University of Victoria.⁶

3.2 Conrad’s Heart of Darkness

Joseph Conrad’s *Heart of Darkness* was studied. The version from the Gutenberg digitization project⁷ used the words, “misunderstanding”, “superciliousness”, “incomprehensible”, “unextinguishable” and “trustworthiness.” This produced a plagiarex of *UAdCWwBvuwqK8Ik9l9eVeA*. A second version in 3 chapters showed the same signature after the text was assembled into one file.⁸

4 Website Revisions

Sina.com and *cnn.com* were studied for significant changes in text. The result was an understanding of when to save a site as a new version. This method is more accurate at detecting significant changes than MD5 alone.

⁴<http://etext.lib.virginia.edu/cgi-bin/browse-mixed?id=ShaHaQ1&tag=public&-images=images/modeng&data=/lv1/Archive/eng-parsed>

⁵<http://etext.lib.virginia.edu/cgi-bin/browse-mixed?id=ShaHaF1&tag=public&-images=images/modeng&data=/lv1/Archive/eng-parsed>

⁶<http://ise.uvic.ca/Annex/DraftTxt/Ham/Ham.F>

⁷<http://www.novelguide.com/heartofdarkness/hdark11.txt>

⁸<http://www.bibliomania.com/0/0/15/28>

4.1 Sina.com

The Chinese web site sina.com⁹ was viewed twice during one day to detect changes. The web site was read by the plagiarex algorithm twice on November 19, 2004. The North American version, *home.sina.com* was studied. An expansion of the meaning of “word” is necessary. For non-phonetic languages, each character is assumed to be a word. This is not true in Chinese, but works well in this context. Using the Simplified Chinese GB2312 font of *sina.com*, the highest order characters that are not whitespace or punctuation are least likely to appear on this site, according to a frequency graph. These characters help to find distinguishing features in the articles. Under Unicode, a different character order changes the plagiarex. With a proper mapping to Unicode, this can be resolved, and the Unicode character set can be used alternatively.

At first, the page gives a plagiarex of *LXFoyhY844cmLaZ2WpOnGQ*. Two hours later, the news changed, creating a plagiarex of *kYXx7AiTG-Uq99zLlGYjEKA*. The text on the page changed, even though an English-reader may not distinguish the two versions.

4.2 Saving Diffs and cnn.com

A *diff*¹⁰ is a record of differences between two files. Given two files that may differ by a few lines of text, the *diff* between the two files might be quite small. A coder might save the original and the *diff*, but throw away the final version. Using *patch*, the final version is recreated in its prior form. This saves disk space in the case of many versions that differ only by a few lines.

One example of saving disk space is with CNN’s web site. Using MD5, the consecutive web pages from CNN look different. The extent of change is not known from the MD5 signature. Comparing the plagiarex of two consecutive snapshots, only a non-consequential HTML comment appears to change. This is where the plagiarex is useful. It shows that the files do not differ in their plagiarex signature. The page is a candidate for saving a *diff*. Here is the command.

```
diff -Naur cnn_1.txt cnn_2.txt > cnn_diff_12.txt
```

cnn_1.txt and *cnn_2.txt* are 58kB long, while the patch, *cnn_diff_12.txt*, is only 820 bytes. This is a significant savings of disk space, while still

⁹<http://home.sina.com>

¹⁰<http://www.cpqlinux.com/patch.html>

enhancing the archive. It is safe to delete `cnn_2.txt` and hold the `plagiarex` values and patch for future use. If the patch grows too large after subsequent downloads, the entire file must be saved again.

To return `cnn_2.txt` to its original state, run the following code.

```
patch -o cnn_2.txt cnn_1.txt cnn_diff_12.txt
```

With minimal use of resources, a record of all changes to a file are saved.

5 Protecting Against Infinite Loops

Crawlers exposed to infinite loops through a series of web pages in directories can overuse resources when dealing with a poorly designed site. In one instance 16,000 nearly identical local directories were created through the Humboldt County web page¹¹. A straight MD5 of the page showed a changing result for each seemingly new page, because the new web pages differed by a small amount. The `plagiarex` MD5, though, showed a constant value of `bpNwvsp4Gf50oihGc+ofLg`. Three key words did not change. They were “modified”, “of” and “at.” Showing the same 3 words on a web page is a sign of machine-readable redundancy. This is an excellent measure by which to stop a crawl.

6 Future Directions

It is important to extend this method to non-English texts, since much of the web and literature is not written in English. To add a language, the character set and subset of allowable characters must be defined. A simple word hierarchy must be documented. Once documented, different programmers should arrive at the same `Plagiarex` values, given the same source. It is important to define a word and delete whitespace and special characters. Allowance, too, should be made for documents with a mix of character sets and Unicode.

Fine tuning of this method leads to more and less sensitive signatures. The default is a five word list. It is acceptable to save `plagiarex(6)` or `plagiarex(1)`, a signature from a six or one word list. The default is `plagiarex(5)`.

¹¹<http://www.co.humboldt.ca.us>

7 Conclusion

The plagiarex MD5 can be used in a variety of applications, including detection of editing of a written work. Two documents with the same plagiarex are likely to be equivalent, differing only by small editorial changes. This was shown to work on texts of Shakespeare, Conrad, CNN and Sina.

This document is written in L^AT_EX.