Plagiarism Detection and Document Chunking Methods

Máté Pataki
Computer and Automation Research Institute, Hungarian Academy of Sciences
MTA SZTAKI, DSD H-1111 Budapest XI. Lagymanyosi u. 11.
+36-1-279-6204
Mate.Pataki@sztaki.hu

ABSTRACT

This paper describes the tests made on chunking methods used for plagiarism detection. The result of the tests makes it possible to decide on the best fitting chunking method for a given application. For example, overlapping word chunking is good for a grammar analyzer or for small databases, sentence chunking suits best for finding quoted texts, hashed breakpoint chunking is the fastest method therefore advisable for search in big set of documents, or if more reliability is needed overlapping hashed breakpoint chunking can be used as well.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]

General Terms

Algorithms

Keywords

plagiarism, similarity detection, chunking, text processing

1. INTRODUCTION

With the evolution of the computer technology/science the process of creating written essays is easier than ever and their publication on the Internet made it fast and cheap as well. So the largest ever seen collection of intellectual works, the Worldwide Web came into being. On the other side digital data storage extremely simplifies the copying of these essays or parts of them, therefore it simplifies plagiarism. Copies, fakes and idea-stealing can be found in many fields of our life. For example for both students and researchers there is an extremely comfortable way for avoiding the hard work of writing their own papers. Sometimes it is not enough to know that a given document is not the product of a particular person, but it is also necessary to prove it, and that can only be done by comparing the questioned document to the original one. To find the original work is nearly impossible without the aid of a computer and that is the reason why more and more plagiarism search engines appear on the Internet these days.

2. RELATED WORK

Plagiarism.org [1] and EVE [2] compare documents to those found on the Internet. InteriGuard System [3] compares uploaded documents to pervious uploaded ones. There are also systems working in a different way. CopyCatch System [4] does not compare documents to a database; it compares several uploaded documents to each other. Glatt Plagiarism Screening Program [5] tries to identify the style of the writer and compare it to others. The most common use of the similarity detection technology is of course to search for plagiarism, but there are many more other uses as well: searching for web pages that store copyrighted documents without permission; searching for documents on a similar theme; locating different versions of the same documents and ordering them; displaying the changes in the development of a document; filtering out identical or very similar documents in a search engine; display only the difference in the documents, so the matching parts won't be displayed twice; quick search for quotations in a big set of documents etc.

3. THE TEST SCENARIO

The heart of the similarity search engine is the chunking method used to chunk the given text into smaller peaces. When comparing documents to each other only these chunks, or their so-called compressed fingerprints, will be compared to determine, how many common parts the documents have. That is the reason why it is so important to have a good chunking method. With the result of these tests it will be possible to build a similarity search engine that best fits the given circumstances. The most important aspects of the methods are (see also Table 1.): speed of the chunking (tested on a big set of documents), size of the database needed (directly proportional to the number of chunks the method generates), the ability to adapt to a specific field using the parameters of the method (called tuning or fine-tuning), and of course the ability to find overlaps. The last one was tested on documents with known content and carefully chosen predefined overlapping. Four chunking methods were tested:

3.1 Sentence Chunking

Sentence chunking seems to be the most obvious method for chunking a text, but it is not that easy. The document can be chunked at the dot, exclamation and question mark (. ! ?) signs, but in some languages, like in Hungarian, the chunks will be long. The comma and semicolon (, ;) signs can be also chunk ends, but then the chunks could get too short. A test involving the RFC [6] documents in English showed an average length of 13 words without and only 3,5 with the use of semicolon sign. Another important attribute is that sentence chunking does not make difference between same sentences in different context, which can be an advantage or a disadvantage as well, depending on the use.

3.2 Hashed Breakpoint Chunking

Hashed breakpoint chunking is the most promising method for chunking big amount of data. The chunking procedure is simple: every word has its own hash value (for example the sum of ASCII numbers in the word), and words with a hash value dividable with parameter *n* are chunk ends. The value for *n* has to be chosen carefully. As shown in Figure 1. it depends highly on the language of the document.

WWW 2003, May 20-24, 2003, Budapest, Hungary.
ACM xxx.

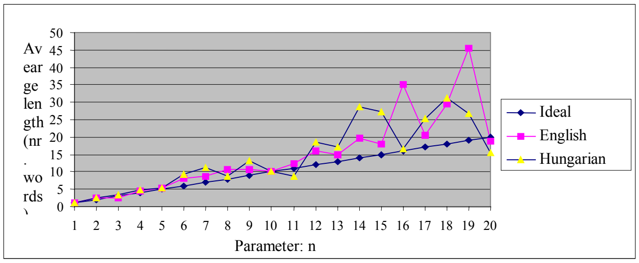

Figure 1. Average chunk lengths for English an Hungarian texts for hashed breakpoint chunking

The tests have shown that where the average is much higher than the ideal, only very long similarities can be identified. It is interesting to compare Figure 1. to Figure 2. For example 16 for parameter *n* would be a good choice for these Hungarian texts but not for the English ones.In Figure 2. the alternate lines show that the same document-pairs with other parameters show up different similarities.

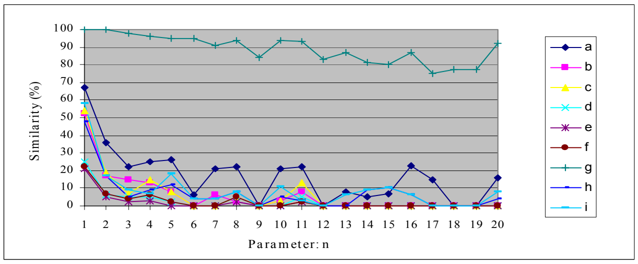

Figure 2. Similarities found in Hungarian document-pairs with hashed breakpoint chunking

Hashed breakpoint chunking is fast and produces a relatively small number of chunks, but it is not reliable if there is no information about the language of documents prior deciding on the parameter *n*.

3.3 Overlapping Word Chunking

In case of overlapped word chunking a chunk begins at every word and contains the next *n* number of words. The number of chunks is equal to the number of words in the text, which makes this method the worst in size of the needed database, but it has the best reliability in finding overlaps (see results).

3.4 Overlapping Hashed Breakpoint Chunking

A new method called overlapping hashed breakpoint chunking was tried out as well. Here the document is chunked using hashed breakpoint method and using more than one parameter, so the chunks overlap each other.

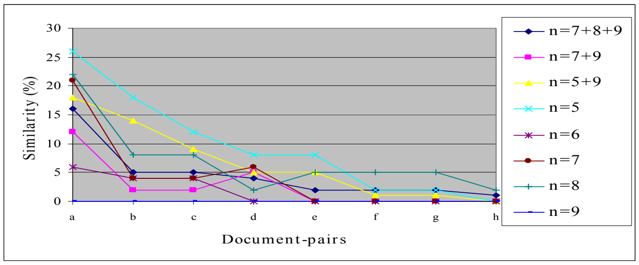

Figure 3. The reliability of Hashed breakpoint chunking compared to overlapping hashed breakpoint chunking

As it can be seen in Figure 3. the parameter 9 shows no difference at all for any of the given document-pairs, but with the use of more parameters together the similarities could always be detected (for example 5+9 means a chunking with parameter 5 and 9 as well).

4. RESULTS

It greatly depends on the field itself which method to use for a special purpose. For example an Internet search engine does not need to be so precise but the database should be as small as possible because a very big amount of files will be stored there. On the other hand for a conference the most important thing is the reliability of the search when testing the incoming publications for plagiarism. They can afford a more complex and slow algorithm because they do not need to deal with as much data as an Internet search engine. The results can be seen in Table 1. (-- means poor and ++ means excellent).

Table 1. Chunking methods compared to each other

	overlapping word	sentence	hashed breakpoint	overlapping hashed breakpoint
chunking speed	-	+ +	+ +	+
size of the database	- -	+	+ +	-
(fine)tuning	+	- -	+ +	+ +
reliability to find overlapping	+ +	0	-	+
one special field it could be used	grammar analyser	finding quoted text	similarty search on the Internet	more reliable similarty search on the Internet

5. ACKNOWLEDGMENTS

My thanks to Arkady Zaslavsky and Krisztián Mosnostori for supervising me during the pervious years. Thanks to Gábor Hodász for the help in the project. [7]

6. REFERENCES

[1] Plagiarism.org, the Internet plagiarism detection service for authors & education. http://www.plagiarism.org , 1999.

[2] EVE Plagiarism Detection System. http://www.canexus.com

[3] InteriGuard System. URL http://www.integriguard.com, 2001. http://HowOriginal.com, http://PaperBin.com

[4] CopyCatch System. http://www.copycatch.freeserve.co.uk

[5] Glatt Plagiarism Screening Program. http://www.plagiarism.com/screen.id.htm, 1999.

[6] Request For Comments: http://www.rfc-editor.org

[7] Computational Science – ICCS 2002, International Conference Amsterdam, Comparison of Overlap Detection Techniques, Krisztián Monostori, Raphael Finkel, Arkady Zaslavsky, Gábor Hodász, Máté Pataki, 10 pages (pp51-60)