PPChecker: Plagiarism Pattern Checker in Document Copy Detection

NamOh Kang,[1] Alexander Gelbukh,[2] SangYong Han [1+]

[1] Chung-Ang University, Korea
kang@archi.cse.cau.ac.kr, hansy@cau.ac.kr

[2] National Polytechnic Institute, Mexico
www.Gelbukh.com

Abstract. Nowadays, most of documents are produced in digital format, in which they can be easily accessed and copied. Document copy detection is a very important tool for protecting the author's copyright. We present PPChecker, a document copy detection system based on plagiarism pattern checking. PPChecker calculates the amount of data copied from the original document to the query document, based on linguistically-motivated plagiarism patterns. Experiments performed on CISI document collection show that PPChecker produces better decision information for document copy detection than existing systems.

1 Introduction

Availability of Internet and easy access to electronic texts makes it easy for many users to share information and to create new documents. However, it also gives convenient environment to malicious plagiarists. This is a serious problem for information sharing. While it is not resolved it will make authors reluctant to share their documents and will reduce the chances for the users to access valuable information.

For protecting the author's copyright, many kinds of intellectual property protection techniques have been introduced: copy prevention, signature- and content-based copy detection, etc. Copy protection and signature-based copy detection can be very useful to prevent or detect copying of a whole document. However, these techniques have some drawbacks that make it difficult for users to share information and can not prevent partial copying of a document [1].

Huge amount of digital documents is made public day to day in Internet. Most of them are not supported by either copy prevention technique or signature-based copy detection technique. This increases the necessity of content-based copy detection technique. So far, many document copy detection (DCD) systems based on content-based copy detection technique have been introduced, such as COPS, SCAM, CHECK, etc. A typical DCD system registers many original documents, compares them with the query document suspected to be plagiarized, and determines the probability of plagiarism. These systems have an advantage of finding the originality of a

[+] Corresponding author: Sang Yong Han.

total or partial copy of a document. However, most DCD systems mainly focus on checking for the possibility of copying between original documents and a query document. They do not give any evidence of plagiaristic sources to the user. In this paper, we describe the plagiarism pattern checking system (PPChecker) that provides evidence information for plagiarism-like style.

The paper is organized as follows. In Section 2, we present some related work. Section 3 explains the system design components used in PPChecker. In section 4, we describe the architecture of PPChecker. In section 5, the experimental result of the system performance is shown. Finally, Section 6 draws conclusion and depicts the future work.

2 Related Work and Motivation

2.1 Content-Based DCD Systems

Document copy detection has been actively researched since 1990s. Many systems for document copy detection have been introduced, such as COPS, SCAM, CHECK, SSK, MDR, etc.

COPS [2] was developed in frame of the Stanford Digital Library Project. It performs comparison between the registered documents and a given query document by a sentence unit. COPS shows very good result in comparing exactly equal sentences, but it can not detect partial sentence overlaps. Shivakumar *et al*. developed SCAM [1] to improve COPS. SCAM uses document word frequency to detect copying. It can find partial overlaps, but comparison of documents sharing many words misleads it.

CHECK [3] extracts structural information and keywords from documents and uses them to check the overlap. It is limited to structured documents. Semantic Sequence Kernel (SSK) [4] first finds out semantic sequences in documents and then uses a kernel function to calculate their similarity. It is good on comparison between non-reworded documents.

Match Detect Retrieval (MDR) system [5] uses string-matching algorithms based on suffix trees to identify the overlap between a suspicious document and candidate documents. It is very powerful for finding exact copy. However, constructing suffix tree for a suspicious document is very expensive, and this system is very weak at detecting modified documents.

WCopyfind [6] uses phrases with six or more words as a comparing unit. It counts the number of words from matching phrases and calculates plagiarism rate as a ratio of the number of matching words and the total number of words in the document. WCopyfind could find a partial overlap, but the user should set an adequate word number in a phrase.

2.2 Plagiarism Patterns

Plagiarism pattern research has not reached its maturity. It is hard to find a scientific classification of plagiarism, except for Karen Fuallam *et al*.'s plagiarism patterns and their variants [7].

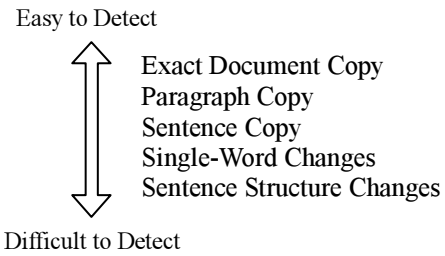

Fig. 1. The plagiarism patterns and their levels of sophistication.

Figure 1 illustrates that it is easier to find an exact copy of the units such as document, paragraph, or sentence than to find the subtle forms of them such as word changes and structure changes.

So far, many DCD systems have been focused at detecting the overlap between documents using various comparing units. However, they do not consider plagiarism patterns. In this paper, we introduce a DCD system based on plagiarism patterns.

3 System Design Components

Comparing unit (chunking unit), overlap measure function, and plagiarism decision function crucially affect the performance of a DCD system. In this section, we consider each of these factors in our system PPChecker.

3.1 Comparison Unit

A DCD system breaks documents down into comparing units (chunking units) for checking the possibility of copying. There are many options to choose comparing unit, such as sentence, paragraph, word, or whole document. For example, COPS uses sentences, SCAM uses words, and CHECK uses paragraphs as comparing unit.

The adopted comparing units affect the accuracy of a DCD system. Selecting large comparing units decreases the necessary number of comparisons but makes it difficult to find partial copy. On the other hand, smaller comparing units increase the number of comparisons and therefore reduce the system's speed. However, it is easy to catch partial copying of the information based on the local similarity.

In this paper, we select sentence as the comparison unit. Sentences form paragraphs or the whole document. Comparison of sentences is a good metric to calculate local similarity. The goal of PPChecker is to provide the user with plagiarism pattern information of the query document. Karen Fullam's research [7] also shows that sentence is a good unit to extract plagiarism pattern information on query document.

3.2 Overlap Measure Function

Overlap measure function is used to determine copying information of the comparing units extracted from documents. Traditionally, many DCD systems use vector space model or cosine similarity model. It is no problem to calculate the similarity between two objects, but it is not enough to calculate the degree of copy. For example, consider the following sentences:

Sentence 1: "A B C D E"	Sentence 2: "A B C D F"
Sentence 3: "G H"	Sentence 4: "G H"

The overlap between sentences 1 and 2 is 4 out of 5 words. Sentence 3 and 4, have a perfect overlap of two items. The degree of such overlap is very important to detect copying: the larger overlap the more evidence of plagiarism. The overlap between the sentences 1 and 2 is two times greater than the overlap between of sentence 3 and sentence 4. So, in the former case overlapping value should be higher than in the latter case. However, the sentences 1 and 2 give 0.8 and the sentences 3 and 4 give 1.0 in cosine similarity. Moreover, the cosine similarity can not give any information of plagiarism. In this research, we suggest the overlap measure function which can quantify the overlap between comparing units and give information about plagiarism.

Let S_o is a part of the original document and S_c of the query document. The similarity $Sim(S_o, S_c)$ can be calculated as follows.

$S_o = \{w_1, w_2, w_3 \ldots\ldots\ w_n\}$, $S_c = \{w_1, w_2, w_3 \ldots\ldots\ w_m\}$

$Comm(S_o, S_c) = S_o \cap S_c$, $Diff(S_o, S_c) = S_o - S_c$

$Syn(w) =$ {The synonym of w}

$SynWord(S_o, S_c) = \{w_i \mid w_i \in Diff(S_c, S_o) \cap Syn(w_i) \in S_o\}$

$$WordOverlap(S_o, S_c) = \frac{|S_o|}{|Comm(S_o, S_c)| + \alpha \times |SynWord(S_o, S_c)|}$$, α is weight value

$$SizeOverlap(S_o, S_c) = \sqrt{|Diff(S_o, S_c)| + |Diff(S_c, S_o)|}$$

$$Sim(S_o, S_c) = |S_o| \times \left(\frac{1}{e^{WordOverlap(S_o, S_c)-1} + SizeOverlap(S_o, S_c)}\right)$$

Calculation of $Sim(S_o, S_c)$ gives not only similarity between S_o and S_c but also the plagiarism information. The following table 1 shows how to decide plagiarism patterns.

The proposed overlap measure function can not distinguish between an exact copy and changing structure of a sentence. For distinguishing them, it is necessary to check the word order in sentences.

Table 1. Plagiarism patterns and their decision parameters.

Plagiarism pattern	Decision parameters	
Sentence copy exactly	$WordOverLap(S_o, S_c) = 1$,	$SizeOverlap(S_o, S_c) = 0$
Word insertion	$SizeOverlap(S_o, S_c) \neq 0$,	$Diff(S_o, S_c) > 1$
Word removal	$SizeOverlap(S_o, S_c) \neq 0$,	$Diff(S_c, S_o) > 1$
Changing word	$1 < WordOverLap(S_o, S_c) < \infty$,	$SizeOverlap(S_o, S_c) = 0$
Changing sentence	$WordOverLap(S_o, S_c) = 1$,	$SizeOverlap(S_o, S_c) = 0$

3.3 Plagiarism Detection

Sentence level overlap and plagiarism pattern information can be calculated by using overlap measure function. However, the decision on whether a document is copied or not is made basing on its global copy information. This information is very useful to generate ranking information to be supplied to the user when many documents are checked for the possibility of copying. In PPChecker, the global copying degree of the query document is calculated by gathering sentence level information in paragraph or document unit. If all sentences of a paragraph or a document pass a given threshold value, some weight value is used to calculate the global degree of copying.

4 The Architecture of PPChecker

The architecture of PPChecker is shown in Figure 2.

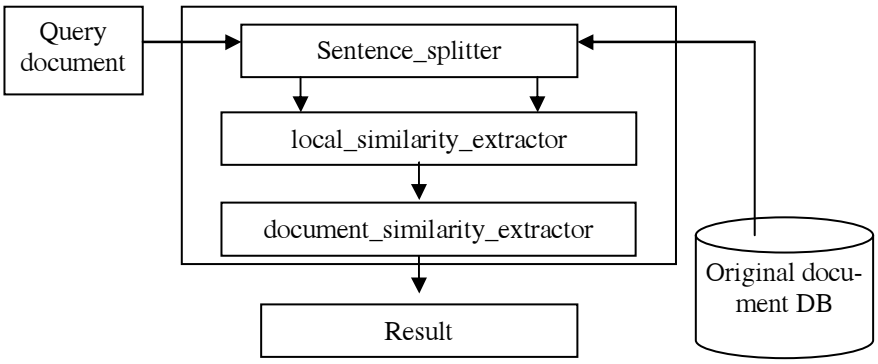

Fig. 2. The architecture of PPChecker

All original documents are stored in the document data base. When the query document is received, the system divides it and the original documents into comparing units—sentences. These sentences are then used to calculate the overlap and the

plagiarism information in *local_similarity_extractor* function (see below) by using the overlap measure function defined in Section 3.2. The extracted information is used to calculate the degree of copying from each other in original documents, and the ordered information is supplied to the user. The algorithm used in PPChecker is as follows:

Algorithm 1

Input:

$Document\_DB = \{D_1, D_2, D_3, \ldots\ldots, \quad D_n\}$ and each $D_i = \{S_{i1}, S_{i2}, S_{i3}, \ldots\ldots, \quad S_{im}\}$

$QueryDocument = \{QS_1, QS_2, QS_3, \ldots\ldots, \quad QS_t\}$

Output:

Decreasing ordered document list in document similarity value

for i = 1 to n

for j = 1 to t

localsimilarity [1..j] = 0;

for k = 1 to m

if $|Comm(S_{ik}, QS_j)| \geq \frac{|S_{ik}|}{2}$ then // $\frac{|S_{ik}|}{2}$ is selected threshold value

localsimilarity [j] = max {localsimilarity [j], $Sim(S_{ik}, QS_j)$ }

end

end

if all similarities in document is over than threshold value

documentsimilarity [i]= $D\_w \times \sum_n localsimilarity[n]$

else if all similarities in a paragraph is over than threshold value

documentsimilarity [i]= $\sum_l localsimilarity[l] + P\_w \times \sum_m localsimilarity[m]$

else

documentsimilarity [i]= $\sum_l localsimilarity[l]$

// P_w and D_w are Paragraph_weight and Document_weight, respectively.

end

return sort(documentsimilarity)

5 Implementation and Experiment

PPChecker was implemented under Windows XP using the language *C#*. It uses Porter stemmer for stemming and WordNet for finding synonyms. To evaluate the system, we use the CISI document collection set. For the experiment, we generated the test document set from CISI as follows.

1. 11 relevant documents related to a specific query were selected from CISI set.
2. One document was selected as the original document. The other 10 documents were selected as candidate documents for plagiarism test.

3. Some text extracted from the original document was transformed (exact copy, changing synonyms, changing sentence structure) and was inserted into the candidate documents for plagiarism detection to make a plagiarized document.
4. The plagiarized documents were returned into the CISI document set. Selected original document was removed from the CISI document set and became the query document.

As a baseline, we implemented a DCD system based on word similarity of the whole document (WD_System) and of a sentence (WS_System). We measured quality in terms of R-precision; R = 10.

Table 2. Copy detection test (R = 10).

		WD_System	WS_System	**PPChecker**
Test 1	Exact copy	2	6	**8**
	Synonym	2	6	**8**
	Structure change	1	**5**	4
Test 2	Exact copy	1	7	**9**
	Synonym	1	6	**7**
	Structure change	1	**3**	**3**
Test 3	Exact copy	1	6	**8**
	Synonym	0	4	**7**
	Structure change	0	3	**4**

In case of exact copy detection and exchanging synonyms, PPChecker give better results than other systems. For changed sentence structure, PPChecker and WS_System give similar results. This shows that in case of checking for changed structure of sentence, information on the words and the synonyms is not necessary to detect the possibility of copying.

6 Discussion and Future Work

The experimental results show that PPChecker produces more precise results in exact copy detection and copy with changing for synonyms. PPChecker's overlap measure function is more useful than normalized comparison value such as cosine similarity. Consideration of plagiarism pattern information produced in comparison helps to make a more precise decision.

In our experiments, only the documents registered in document database could be used to detect plagiarism. However, nowadays huge amount of documents are open for access in Internet and thus vulnerable to plagiarism. Thus research on using Internet as document database is necessary in the future.

PPChecker uses WordNet for checking word synonymy. If we use a domain-specific ontology for this, precision could be improved. Finally, research on plagiarism patterns is to be continued due to its crucial importance for document copy detection.

References

1. Shivakumar, N. and Garcia-Monlina, H. “SCAM: A Copy Detection Mechanisms for Digital Documents.” In Proceedings of International Conference on Theory and Practice of Digital Libraries, Austin, Texas. June 1995.
2. Brin, S., Davis, J., and Garcia-Molina, H. "Copy Detection Mechanisms for Digital Documents." In Proceedings of ACM SIGMOD Annual Conference, San Jose, CA, May 1995.
3. Si, A., Leong, H., and Lau, R. "CHECK: A Document Plagiarism Detection System." In Proceedings of ACM Symposium for Applied Computing, pp. 70-77, Feb 1997.
4. Bao Jun-Peng, Shen Jun-Yi, Liu Xiao-Dong, Liu Hai-Yan, Zhang Xiao-Di. "Document Copy Detection Based On Kernel Method,” In 2003 International Conference on Natural Language Processing and Knowledge Engineering Proceedings.
5. Krisztian Monostori, Arkady Zaslavsky, Heinz Schmidt “Document Overlap Detection System for Distributed Digital Libraries”, Proceedings of the fifth ACM conference on Digital libraries, 2000, pp. 226 – 227.
6. Louis Bloomfield, http://plagiarism.phys.virginia.edu, The Plagiarism Resource Site Charlottesville, Virginia.
7. Karen Fullam, Jisun Park "Improvements for Scalable and Accurate Plagiarism Detection in Digital Documents" 2002.
8. Narayanan Shivakumar, Hector Garcia-Molina "Building a Scalable and Accurate Copy Detection Mechanism" 1st ACM International Conference on Digital Libraries (DL'96), March. 1996. pp. 160-168.
9. Finkel, R., Zaslavsky, A. Monostori, K., and Schmidt, H. "Signature Extraction for Overlap Detection in Documents." In Proceedings of Australasian Computer Science Conference, 2002.