EPCI: Extracting Potentially Copyright Infringement Texts from the Web

Takashi Tashiro[1], Takanori Ueda[1], Taisuke Hori[1], Yu Hirate[1] and Hayato Yamana[1,2]
[1]Computer Science Div., Waseda University, 3-4-1 Okubo Shinjuku-ku, Tokyo, Japan
[2]National Institute of Infomatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{ttashiro, ueda}@yama.info.waseda.ac.jp,mail_to_hori@toki.waseda.jp,
{hirate,yamana}@yama.info.waseda.ac.jp

ABSTRACT
In this paper, we propose a new system extracting potentially copyright infringement texts from the Web, called EPCI. EPCI extracts them in the following way: (1) generating a set of queries based on a given copyright reserved seed-text, (2) putting every query to search engine API, (3) gathering the search result Web pages from high ranking until the similarity between the given seed-text and the search result pages becomes less than a given threshold value, and (4) merging all the gathered pages, then re-ranking them in the order of their similarity. Our experimental result using 40 seed-texts shows that EPCI is able to extract 132 potentially copyright infringement Web pages per a given copyright reserved seed-text with 94% precision in average.

Categories and Subject Descriptors:
H.3.3 [INFORMATION STORAGE AND RETRIEVAL]: – Information Search and Retrieval

General Terms: Experimentation

Keywords: Information Retrieval, Copy Detection

1. INTRODUCTION
In recent years, due to widespread distribution tools such as Blog and Wiki, copyright infringement texts have increased rapidly on the Web. However, we hardly detect such texts from the Web manually, because the size of the Web is too huge and there also exist many copyright reserved texts.

The conventional approach to solve this kind of problem uses "register server [3]" in which copyright reserved texts are stored. Writers store their works into a document database on the register server. Then the register server checks the document whether it has same sentences as those in the register server or not. Other important works are [1][2][4] that extract similar texts from the Web by using search engine APIs such as Google API. They detect plagiarism in an input text such as student reports by extracting similar texts from the Web.

Although variety of methods[1][2][3][4] have been proposed, none of them is applied to extract potentially copyright infringement texts from the Web as many as possible in order to increase recall. As for the method using register server[3], we should store all the Web pages into the register server, however, it may result in impossible because the size of the Web is huge. Thus, the best way is to use search engines such as Google, Yahoo! or MSN as backend. The conventional methods[1][2][4] using such search engines, however, are not able to extract potentially copyright infringement texts from the Web as many as possible, because they just aim to detect the existence of the similar texts in the Web. Thus, they are able to extract a few similar texts from the Web that results in low recall.

To solve the problem, we propose a new system for extracting potentially copyright infringement texts from the Web called EPCI. Here, we define potentially copyright infringement texts as the texts which may include similar phrases of copyright reserved texts that a user has. We call the copyright reserved texts as "seed-texts" in this paper. Since the judgment whether a text is copyright infringement or regal use is difficult, EPCI provides ranked Web pages in the descending order of similarity with a given seed-text. Our contribution is to propose both a new similarity based ranking scheme and a new query generating scheme to satisfy (1) enabling narrowing down search to exclude unrelated pages and (2) enabling to include in the search results some modified pages based on a seed-text.

WWW 2007, May 8--12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005..

2. THE ARCHTECTURE OF EPCI
EPCI consists of Chunker, Query Generator, Candidate Page Getter and Similarity Analyzer as shown in Figure 1. When EPCI receives a seed-text, Chunker divides it into a series of chunks. Then, Query Generator generates queries based on the series of chunks. After that, Candidate Page Getter puts them to search engine API. After receiving N search results from high ranking, Similarity Analyzer calculates the similarity, defined in 2.4, between the seed-text and search result Web pages. While the similarity is higher than a given threshold, Candidate Page Getter gathers next N search results form search engine APIs and hands them to Similarity Analyzer. At last, all gathered pages are re-ranked with their similarity.

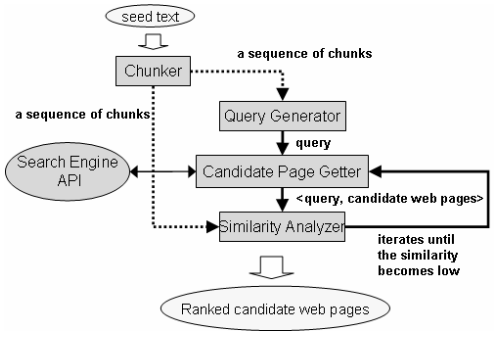

Figure 1 Overview of EPCI

2.1 Chunker
Chunker divides a seed-text into chunks and passes them to Query Generator and Similarity Analyzer. Chunks used in EPCI are

defined as follows. In the case of English text, chunk is defined as two adjacent words not to overlap one another. In the case of Japanese text, chunk is defined as the Japanese phrasal units, called "bunsetsu" in Japanese.

2.2 Query Generator

Query Generator generates a set of queries for search engines based on a sequence of chunks, and passes each query to Candidate Page Getter. The crucial requirements for queries are (1) enabling narrowing down search to exclude unrelated pages and (2) enabling to include in the search results some modified pages based on a seed-text. In order to satisfy both requirements, we adopt n-gram technique to generate a set of queries from a sequence of chunks as shown in Table 1. The generated search queries are sets of adjacent 2-chunks both to narrow down search results and to include similar texts which contain at least adjacent 2-chunks.

Table 1.Example of Chunks and Search Queries using bi-gram

chunks	"fortune comes", "in by", "a merry", "gate"
query 1	"fortune comes in by"
query 2	"in by a merry "
query 3	"a merry gate"

2.3 Candidate Page Getter

Candidate Page Getter receives a search query, and gathers candidate Web pages by using search engine APIs. Candidate Page Getter, after it receives a search query, executes following two processes: (1) putting the received query to search engine API as phrase search, and getting N search results from high ranking where N is pre-defined such as 10, (2) gathering Web pages themselves by using the URLs in the search results.

2.4 Similarity Analyzer

Similarity Analyzer calculates similarity value which quantifies similarity between a part of a Web page and a seed-text. The similarity value is defined by equation (1).

$$Sim(S,C) = \log_2 \left\{ \frac{|LCS(S,C)|}{|S|} + 1 \right\} \quad \cdots (1)$$

where, S indicates a sequence of chunks in seed-text, C indicates a sequence of chunks in a Web page, and $|LCS(S,C)|$ indicates the length of the longest common subsequent chunks between S and C. The longest common subsequent chunks are defined as the longest series of chunks that are included in both S and C as a subsequence.

By using LCS, even if a Web page contains unrelated words or paragraphs to a seed-text such as comments, the similarity value remains in same value.

3. EVALUATION

In this experiment, we used 40 seed-texts in total, 10 each from following 4 categories: English news from http://news.yahoo.com, English lyrics from http://lyrics.astraweb.com, Japanese news from http://news.goo.ne.jp and Japanese lyrics form http://www2.kget.jp. These seed-texts vary in length from 163 to 788 words. In this evaluation, we used Yahoo! JAPAN Web Search API as the backend of EPCI. Then, EPCI extracted potentially copyright infringement pages whose similarity values calculated by equation (1) are more than 0.3. Then, the extracted pages are munually examined their similarity with each given seed-text in order to confirm the precision. Table 2 shows the definition of manualy examined similarity score. The potencially copyright infringement pages should have the score 1, 2 or 3.

Table 2. Definition of Score for Extracted Web Pages

	Percentage of the overlapping between an extracted Web page and the seed-text
Socre3	more than 80%
Score2	30% to 80%
Score1	less than 30%
Score0	0% (Unrelated Web page)

Table 3 shows the extracted Web pages categorized by their score. As shown in Table 3, we have confirmed that EPCI is able to extract potentially copyright infringement Web pages with 94% precision in average. We have also confirmed that EPCI extracts more then 100 potentially copyright infringement pages per a seed-text except in Japanese news category. In Japanese news category, the number of extracted pages is fewer than the number of results in other categories. The reason of this is thought that Japanese news has tendency to be fewer plagiarized.

Moreover, we have confirmed that EPCI is able to extract many parody songs and many commented texts. This shows that EPCI could be robust against changes of words.

Table 3. Extracted Web Pages Categorized by their Score

	Score3	Score2	Score1	Score0	Total
English news	1,658	334	26	28	2,046
English lyrics	1,075	272	16	160	1,523
Japanese news	213	68	19	72	372
Japanese lyrics	1,240	354	36	62	1,692
Total (40seed-texts)	4,186 (74%)	1,028 (18%)	97 (2%)	322 (6%)	5,633 (100%)

4. CONCLUSION

In this paper, we propose EPCI: the system which extracts potentially copyright infringement texts from the Web. Our experimental result shows that EPCI is able to extract 132 copyright infringement Web pages per a given copyright reserved seed-text with 94% precision in average.

5. REFERENCES

[1] A.R. Pereira and N. Ziviani, "Retrieving Similar Documents form the Web," J. of Web Engineering, Vol. 2, No. 4, pp.247-261, 2004.

[2] C. J. Neill and G. Shanmuganthan, "A Web-Enabled Plagiarism Detection Tool," IT Professional, Vol. 6, No. 5, pp. 19-23, 2004.

[3] S. Brin, J. Davis and H. Carcia-Molina, "Copy Detection Mechanisms for Digital Documents," Proc. of the ACM SIGMOD Ann. Conf., pp.398-409, 1995.

[4] S. Niezgoda and T. P. Way, "SNITCH: A Software Tool for Detecting Cut and Paste Plagiarism," Proc. of the 37th SIGCSE Tech. Symp. on Computer Science Education, pp.51-55, 2006.