Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 25 (2016) 372 – 377

Procedia Technology

Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST 2016)

A plagiarism Detection System for Malayalam Text based documents with Full and Partial Copy

Sindhu.L[a*], *Sumam Mary Idicula* [b],

[a]*Department of computer science, College of Engineering, Poonjar, 686582, India*
[b]*Department of Computer Science, Cochin University of science and Technology, 682022, India*

Abstract

Plagiarism or the act of copying some other persons work is increasing, with the availability of a huge amount of digital documents online. The widespread use of the Internet has made the copying of documents very easy. Documents can be copied completely or partially. Many document copy detection systems have been proposed ,but an efficient detection system for Malayalam documents is not available. In this paper, a technique for detecting full and partial copies is proposed. SCAM algorithm is used to find out similar paragraphs and then PPChecker algorithm for determining degree of plagiarism at paragraph level and document level. The system is evaluated with metrics of precision and recall. The system obtained high precision which proves the effectiveness of the proposed method.

Peer-review under responsibility of the organizing committee of RAEREST 2016

Keywords: Plagiarism detection; partial copy detection; plagiarism; SCAM; PPCHECKER ;

1. Introduction

Plagiarism, which is the act of passing off somebody else's original words and ideas as one's own, is seen as a moral offence and often also a legal offence. As more and more information becomes available online, the sheer amount of information for manual investigation becomes overwhelming. Hence, computational methods have been introduced to aid text reuse, authorship and direction identification. This is where automatic plagiarism detection started to gain attention, as it may be able to offer an effective and efficient solution, at a lower economic cost than using human resources. Plagiarism detection in natural languages by statistical or computerized methods has started since the 1990s, with the studies of plagiarism detection methods in digital documents [1], [2]. The last decade witnessed research on plagiarism detection in natural languages employing techniques from information retrieval (IR), natural language processing, computational linguistics, artificial intelligence etc. Many software tools have

Peer-review under responsibility of the organizing committee of RAEREST 2016
doi:10.1016/j.protcy.2016.08.120

been developed for plagiarism detection which are either web-based systems or stand-alone systems [3]. Examples of some tools are EVE2, Plagiarism-Finder, WCopyFind , Turnitin, SafeAssign, COPS , MDR , SCAM , PPCHECKER , CHECK , SNITCH etc. The paper is organized as follows. In section 2, related works especially SCAM and PPCHECKER are discussed. . The proposed methodology is described in section 3. In section 4, experimental setup and the results obtained are discussed. Section 5 concludes the paper with discussions on future works.

2. Related works

COPS [1] performs comparison between the sentences of the query document and original ones. Each sentence is hashed and the hash value is used for detecting a copied sentence. If two documents are found to have common sentences whose number exceeds a threshold then plagiarism is suspected.. This method can detect only full sentence copy but it cannot detect the partial copy of sentences.

WCOPYFIND [3] use phrases having at least six words as checking unit. Plagiarism rate is calculated as a ratio between the number of words from matched phrases and the total words in documents. This method works very well for exact copy but not in the case of changing or deleting some words.

MDR [4] uses suffix tree for matching sentences. This method can detect either partially or completely duplicated sentences. However, building suffix tree for a document is very expensive.

SCAM [2] detects plagiarism by comparing the word frequency occurrences of the query document and the other ones. It is very fast and it can find the partial copied sentences and is efficient if there are significant copy However if there are only some parts which are copied from other documents, SCAM method shows results with less precision.

PPCHECKER [5] uses the local similarity as unit for checking the possibility of copy. This measure is computed at sentence level by comparing a sentence of query document with all sentences of another one. It bases not only on the set of common words but also the synonyms set between two sentences. PPCHECKER can detect both the whole copied sentences and the partial copied sentences affected by some modifications with high precision. However, it is time-consuming and its performance is affected by stop words.

In PAN 2013, PAN 2014 competitions, different methods for copy detection have been developed. The method used is to find seeds or small similar fragments between two documents. Seeding can be done using techniques such as n-grams, bag-of-words etc. Later larger similar text fragments are formed and filtered for the final results [6]. The seed unit is usually a sentence .

2.1. SCAM (Stanford Copy Analysis Mechanism) [2] determines the degree of plagiarism for a document based on a set of common words. Let R be the query document and S be the original document. SCAM computes the closeness set c(R, S) to contain those word w_i that have similar number of occurrences in the two documents. A word wi is in c(R, S) if it satisfies the following condition:

$$\varepsilon - \left(\frac{F_i(R)}{F_i(S)} + \frac{F_i(S)}{F_i(R)}\right)$$

where $\varepsilon=(2^{+},+\infty)$ is constant parameter. The value of ε was set to 2.5 as the best value in practice. $F_i(R)$, $F_i(S)$ are respectively the number of occurrences of w_i in R and S.

$subset(R,S)$, a subset of document S, is calculated as

$$subset(R,S) = \frac{\sum_{wi\,\varepsilon\, c(R,S)} a_i^2 * F_i(R) * F_i(S)}{\sum_{i=1}^{N} a_i^2 * F_i^2(R)}$$

α_i is weighting value of w_i, usually set to 1.

And then, $sim\ R$, the similarity measure between two documents R and S is defined as follows:

$$sim\ R,S = max\{subset\ R,S\ , subset(S,R)\}$$

If sim , exceeds 1, it is set to 1.
This method has advantage in processing time because determining the closeness set is very fast. However, with a fixed value of ε the chance of matching unrelated documents (the false positives) is increased in function of document lengths because relative position between common words has not been considered. We can control the false positives by modifying the value of ε. A low value of ε will decrease false positives but also decrease the ability to detect the minor overlaps.

2.2. PPChecker (Plagiarism Pattern Checker) algorithm [8] compares a sentence in a query document R with a sentence in an original document S. If R has n sentences and S has m sentences, this algorithm will compare $n \times m$ sentence pairs. The plagiarism degree between R and S is computed from the similarity of each pair.

Let S_q be a sentence in the query document R, S_o be a sentence in the original document S, $sim(S_o,S_q)$ denotes their similarity value, $Comm(S_o,S_q)$ set of common words between S_o and S_q, $Diff(S_o,S_q)$ set of words existing in S_o but not in S_q.

So= $[w_1,w_2,\ldots w_k,\ldots.w_n]$, S_q= $[w_1,w_2,\ldots w_l,\ldots..w_m]$

$Comm\ (S_o,S_q) = S_o \cap S_q$

$Diff\ (S_o,S_q) = S_o - S_q$

Syn(w) be the synonym words of w.

$$Synword(s_0,s_q)=\{w_i \mid w_i \epsilon Diff(s_q,s_0) \cap Syn(w_i) \epsilon s_0\}$$

$$\text{WordOverlap}\ (s_0,s_q) = \frac{|S_0|}{|\text{Comm}(S_0,S_q)| + \alpha \text{ x } |\text{SynWord}(S_0,S_q)|}$$

where α is weight value, usually set to 1.

$$\text{SizeOverlap}(S_0, S_q) = \sqrt{|\text{Diff}(S_0, S_q)| + |\text{Diff}(S_0, S_q)|}$$

Similarity value between $So\ and\ Sq$:

$$\text{Sim}(S_0, S_q) = \frac{|S_0|}{e^{WordOverlap(S_0,S_q)-1} + SizeOverlap(S_0,S_q)}$$

Therefore, similarity value between query document R and original document S can be calculated as follows:

$$sim(S,R) = \sum_{i=1}^{x} \text{sim}(S_s, S_{Ri})$$

where $sim(S_s,S_{Ri})$ is the largest similarity value between the sentence R_i in R (S_{Ri}) and a sentence in the original document S (S_s); x denotes the number of pair (S_s,S_{Ri}) such that $Comm\ (S_S,S_{Ri}) > |\ S_S\ | / 2$.
In case of exact copy or exchanging synonyms, PPCHECKER gives better results than other tested systems [8]. However, one of its disadvantages is processing time because it should check all n x m pairs of sentences.

3. Proposed method

PPCHECKER works effectively to detect either partially or completely copied sentences. This is very useful check if some parts of a document are copied from other documents. However, issue of improving the processing time when comparing documents arises, because all pairs of sentences considered. This is tackled by first

identifying paragraphs that have a high possibility of copy. Next degree of copy of the suspicious paragraphs is calculated. They are finally aggregated to calculate the copy degree at document-level. Plagiarist often have the tendency to copy some paragraphs rather than some sentences. Plagiarism at paragraph level involves copying one or more ideas in full. In this paper, the SCAM algorithm is used to find out similar paragraphs and then PPChecker algorithm is used for determining degree of plagiarism at paragraph level and document level.

Let R be the query document, and S be the original document. The proposed method consists of the following steps:

Step 1: Documents R and S are split into paragraphs where $R=\{p_{R1},p_{R2},\ldots.p_{Rk},\ldots.p_{Rn}\}$ and $S=\{p_{S1},p_{S2},\ldots.p_{Sl},\ldots.p_{Sm}\}$

Step 2: SCAM algorithm is incorporated on n*m pairs of paragraphs. The similarity degree of each pair $sim(p_{Ri},p_{Sj})$ will be calculated . For each paragraph p_{Ri} in R, the most similar one in S, matched(p_{Ri}) is found. Only pairs whose similarity degree exceeds given threshold is retained.. Therfore the number of pairs retained for checking plagiarism is reduced to k where ($k \leq n$) .

$G= P_{RS1},P_{RS2},\ldots,P_{RSi},..P_{RSk}$ $P_{RSi} = \{p_{Ri},\text{matched}(p_{Ri})\}$

$$matched(P_{Ri}) = P_{SI} \quad \left| sim\left(P_{Ri},P_{SI}\right) max_j\left(sim\left(P_{Ri},P_{Sj}\right)\right)\right.$$
$$sim\left(P_{Ri},P_{SI}\right) > threshold$$

Step 3: Computing similarity degree at paragraph level

PPCHECKER algorithm is applied for each paragraph pair from G. Here, a paragraph is considered in place of the document. The similarity value sim_i of the pair P_{RSi} is computed as in ppchecker with the difference that two sentences are detected as copied if

$$Comm(S_s,S_{Ri}) > \frac{|S_S|+|S_{Ri}|}{4}$$

Let n_i is the number of copied sentences for $P_{RSi.}$

Step 4: Computing the similarity degree and copy rate date document level.

The similarity between two documents R and S is calculated as

$$sim(R,S) = \sum_{i=1}^{k} \text{sim}_\text{i}$$

and the copy rate of R to S is calculated as

$$rate(R,S) = \frac{\sum_{i=1}^{k} n_i}{|R|}$$

The ***sim()*** value shows the quantity of copied parts while ***rate ()***value indicates the percentages of sentences in the query document R is copied from the original S. Rate presents the ratio between number of copied sentences and number of all sentences in query document.

4. Experimental design and data set

4.1. Data Sets

Malayalam documents collected from Malayalam online newspapers are used for the experiments.

One document is taken as the original document A . Ten documents were created from the original document through various levels of copying from the original document .The rest of each document was randomly selected from another document which contains another topic.. These ten documents generated are the set of *query* documents to compare with the original document. This forms data set D1.

Another document B is selected which belongs to the same topic as the original document. Ten documents were created from the original document through various levels of copying from the original document .The rest of each

document was randomly selected from B which contains same topic.. These ten documents generated are the set of *query* documents to compare with the original document. This forms data set D2.

4.2 Experiments on data set

Initially some preprocessing is done. The documents are split to paragraphs and then to sentences and then to words. Stop words and special characters are removed. The value of ε, α and the threshold are set.

Two sets of documents D1 and D2 were used to evaluate the performance of the proposed method, SCAM and PPChecker. Each document in D1 or D2 is compared to the original document. SCAM checks for words common to both documents for computing the copy degree between them, while PPChecker and the proposed approach checks for copied sentences.

All the three algorithms perform well for direct or word to word copy between documents. Data set D1contains parts of documents from different topics. Data set D2contains parts of documents from same topics. Therefore even though the part of the document is not copied, it may have words in common with the original document. The SCAM algorithm detects high similarity resulting in false positives. The proposed method is able to identify even very small partial copy as small as 10%.

On comparison with SCAM and PPChecker , it is found that the proposed method using features of both the SCAM and PPChecker algorithm, reduces the possibilities of false positives and improves the detection of even small partial copies without compromise in processing time.

5. Conclusions

Plagiarism is a problem to be tackled. Documents may be copied from other documents fully or partially. Plagiarism prevention is the best solution. But since prevention is not effective, plagiarism detection methods are necessary. In this paper , a method for copy detection where copy of part of a document or the entire document can be detected was discussed. The documents are processed at paragraph level before sentence level. Experiments prove the efficiency of the approach in terms of accuracy and of time. The proposed method is better than SCAM in terms of false positives and is also better than PPChecker with respect to processing time.

References

[1] S. Brin, J. Davis, and H. Garcia-Molina, “Copy detection mechanisms for digital documents,” in *Proc. ACMSIGMOD Int. Conf.Manage. Data*, New York, 1995, pp. 398–409. Van der Geer J, Hanraads JAJ, Lupton RA. The art of writing a scientific article. J Sci Commun 2000;163:51-9.

[2] N. Shivakumar and H. Garcia-Molina, “SCAM: A copy detection mechanism for digital documents,” in International Conference in Theory and Practice of Digital Libraries (DL 1995).

[3] Bloomfield, L. 2014. *The Plagiarism Resource Site*. http://plagiarism.bloomfieldmedia.com (last updated 2014)

[4] Bin-Habtoor, A. S, and Zaher, M. A. 2012. *A Survey on Plagiarism Detection Systems*, International Journal of Computer Theory and Engineering Vol. 4, No. 2, April 2012.

[5] NamOh, K., Gelbukh, A., Sang Yong, H. 2006. PPChecker: *Plagiarism Pattern Checker in Document Copy Detection*. In Proceedings of the 9th international conference on Text, Speech and Dialogue, pages 661-667.

[6] Forner, P., Müller, H., Paredes, R., Rosso, P., & Stein, B. (eds). 2013. *Information Access Evaluation Meets Multilinguality, Multimodality, and Visualization*. 4th International Conference of the CLEF Initiative (CLEF 2013), September 2013.

[5] Le-Hong, P., T M H. Nguyen, A. Roussanaly, and T V. Ho. 2008. *A hybrid approach to word segmentation of Vietnamese texts*. Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, p.240-249.

[6] Miguel Sanchez-Perez, Grigori Sidorov, Alexander Gelbukh. 2014. *The Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014*. In: L. Cappellato, N. Ferro, M. Halvey, W. Kraaij (eds.). Notebook for PAN at CLEF 2014. CLEF2014 Working Notes. Sheffield, UK, September 15-18, 2014. CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1180, CEUR-WS.org, 2014, pp. 1004–1011.

[7] Monostori, K,Zaslavsky, A., Schmidt, H. 2000. *Document Overlap Detection System for Distributed Digital Libraries*, In proceedings of the fifth ACM conference on Digital libraries, pp. 226 – 227.

[8] NamOh, K., Gelbukh, A., Sang Yong, H. 2006. PPChecker: *Plagiarism Pattern Checker in Document Copy Detection*. In Proceedings of the 9th international conference on Text, Speech and Dialogue, pages 661-667.

[9] Shivakumar, N., Garcia-Molina, H. 1995.*SCAM: A Copy Detection Mechanism for Digital Document*, International Conference in Theory and Practice of Digital Libraries (DL 1995).

[10] Shivakumar, N., Garcia-Molina H.. 1996. *Building a Scalable and Accurate Copy Detection Mechanism*. 1st ACM International Conference on Digital Libraries (DL'96),pp. 160-168

[11] Si, A., Leong, H., and Lau, R. 1997. *CHECK: A Document Plagiarism Detection System*. In Proceedings of ACM Symposium for Applied Computing, pp. 70-77(Feb 1997).

[12] Sebastian Niezgoda and Thomas P. Way. 2006. *SNITCH: a software tool for detecting cut and paste plagiarism*. In Proceedings of the 37th SIGCSE technical symposium on Computer science education (SIGCSE '06). ACM, New York, NY, USA, 51-55. DOI=http://dx.doi.org/10.1145/1121341.1121359
[13] Eve 2: http://www.canexus.com/
[14] Plagarism-Finder: http://www.m4-software.com/en-index.htm (last update 2004)
[15] TurnItin : http://turnitin.com/
[16] SafeAssign http://www.safeassign.com/