A CONCEPTION OF A PLAGIARISM DETECTION TOOL FOR PROCESSING TEMPLATE-BASED DOCUMENTS

Romans Lukashenko, Alla Anohina, Janis Grundspenkis
Riga Technical University
Kalku Street 1, Riga, LV-1658, Latvia
E-mail: {LREXPress@inbox.lv, alla.anohina@cs.rtu.lv, janis.grundspenkis@cs.rtu.lv}

KEYWORDS

Plagiarism, plagiarism detection tools, template-based documents.

ABSTRACT

Plagiarism detection in template–based documents is an actual task for educational environments, because many teachers prefer to make templates of submissions for their learning courses. Nowadays available plagiarism detection tools are inefficient in detecting plagiarism in template-based documents due to several reasons. In this paper the conception of a new plagiarism detection tool for processing especially template-based documents is presented and applicability of such a tool for academia purposes is discussed as well.

INTRODUCTION

Modern societies evolve in the information age. Information technologies, from one hand, make our life easier, but, from the other hand, create a set of problems as well. Availability of digital documents (especially, through easy access to the Web) and telecommunications in general provides good conditions for the prosperity of such social illness as plagiarism (Lukashenko et. al. 2007).

Plagiarism can be defined as turning of someone else's work as your own without reference to the original source (Maurer et. al. 2006). Nowadays plagiarism as theft of intellectual property has turned into a serious problem for publishers, researchers and educators (Maurer et. al. 2006). In educational environments plagiarism is a significant breach for three reasons. Firstly, this phenomenon is in contradiction to the process of learning which demands from a learner to take certain intellectual and physical efforts in order to acquire knowledge and skills necessary for the further social and professional activity. Secondly, plagiarism reduces the value of a qualification conferred by the educational institution. Thirdly, it demotivates other students to work independently and to put efforts to learning in case of impunity of plagiarism.

Many different software tools for automated plagiarism detection are already developed and used, for example, Turnitin, Eve2, CopyCatchGold, WordCheck, Glatt, Moss, JPlag (Maurer et. al. 2006, Delvin 2002, Lancaster and Culwin 2000, Lancaster and Culwin 2005, Neill and Shanmuganthan 2004, The University 2003). In general they provide excellent service for detecting matching text in documents (The University 2003). However the analysis of the mentioned plagiarism detection tools shows that one of the serious identified problems is tools inability to detect plagiarism correctly in a set of documents which are created using the same document template. To overcome this drawback we have introduced the conception of a new plagiarism detection tool for processing especially template-based documents.

The remainder of this paper is organized as follows. In the next section description of the existing plagiarism detection tools is given. Afterwards our experience of the usage of the document template is described. Then the conception of the plagiarism detection tool for processing template-based documents is presented. Applicability of such a tool is discussed as well. The final section is devoted to conclusions and directions for future work.

PLAGIARISM DETECTION TOOLS

In accordance with (Lancaster and Culwin 2005) "plagiarism detection tools are programs that compare a document with possible sources in order to identify similarity and so discover submissions that might be plagiarized". Figure 1 illustrates a common work scheme of a plagiarism detection tool. A submitted document is an input of the similarities detection block which tries to find similarities between the document and possible sources. The possible sources can be located either intra-corpally, for example, local DB of documents, or extra-corpally, for example, sources in the Internet. The results of similarities detection are presented as percentage of plagiarism in the document.

The abovementioned working principles are implemented in a number of plagiarism detection tools listed in Introduction. In addition, Internet search engines also can be viewed as alternative tools to detect suspected plagiarism.

Documents' content analysis in plagiarism detection tools is based on semantical methods (Aslam and Frost

2003, Brin et. al. 1995) or statistical methods (Brin et. al. 1995, Lancaster and Culwin 2004). The most widely used are statistical methods, because there is no need to understand the meaning of the document as it is required by semantical methods. One of the popular purely statistical methods is N-gram method (Brin et. al. 1995, Tan et. al. 2003), where a document content is characterized with sequences of N consecutive grams (for a text document the gram can be a letter, a word, a sentence, etc.). The method is based on the comparison of selected grams from one document with appropriate grams from the other document to find identical regions in two documents and to make a decision about the degree of similarity between the documents.

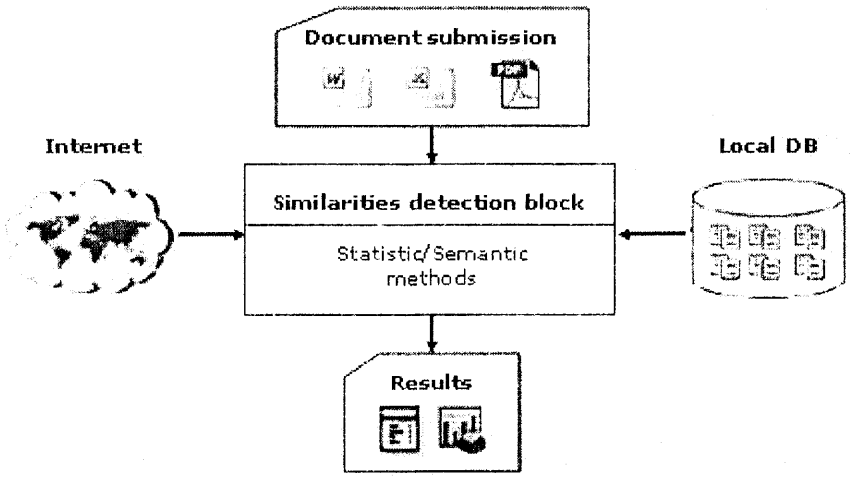

Figure 1. Plagiarism detection process (adapted from (Aslam and Frost 2003)).

Plagiarism detection tools operate mainly on free text or source code. Tools also can be used successfully to find similarity in spreadsheets, diagrams, scientific experiments, music or any other non-textual corpora (Lancaster and Culwin 2005).

In spite of the fact, that plagiarism detection tools in general provide good service for detecting matching between documents, their usage in educational environments can be limited in some cases. It is known that many teachers prefer to make templates of submissions for their learning courses to unify formatting rules and to provide easy checking of submissions. However, analysis of plagiarism detection tools shows that tools are inefficient in processing documents which are created using a template, for example, processing a set of students' course works which are created using a course work's template prepared by a teacher. We found that there are two problems in processing template-based documents.

Firstly, plagiarism detection tools check a whole document and as a result legally identical text areas, for example, tasks statements which are the same in all documents are identified as plagiarized regions. It distorts the overall results of plagiarism detection process and makes difficult to reason about real amount of plagiarism in the document.

Secondly, plagiarism detection tools compare all parts of one document with all parts within another document for mutual similarities and as a result semantically unrelated parts of two documents, for example, two solutions of absolutely different tasks, are processed unnecessarily. It increases the overall computational time of document processing.

Taking into account the described problems we have developed a conception of the plagiarism detection tool for processing template-based documents. However, before we will start its discussion, let's consider our experience of the usage of the submission template.

EXPERIENCE OF THE USAGE OF THE DOCUMENT TEMPLATE

A document template is the pre-designed layout which provides consistent format and content of data put into it. In educational environments templates are used for standardization and easy checking of students' submissions. In our Riga Technical University we have an experience of template usage within the learning course "Fundamentals of artificial intelligence" for third year students of bachelor study programs at the Faculty of Computer Science and Information Technology. The obligatory constituent part of the mentioned course is the development of the course work which students prepare independently outside the university. Ten tasks are included in the course work covering the main topics of the course: representation of the state space, uninformed and informed search, modeling of two-person games and knowledge representation. Until recently students received only statements of the tasks, individual

parameters and guidelines for the development of the course work. Taking into account that the number of students reaches 250 persons on average every year, several serious problems were related with the checking of students' submissions. Firstly, students put into their works much unnecessary information (text and graphics) which is not significant for the completion of the tasks. It, certainly, made the searching of the task solution difficult for the teacher, as well as extended the total time for checking of submissions. Secondly, course works had great differences in formatting that made difficult their assessment. For example, in tasks, where the solution can be acquired by performing several intermediate steps, one student could show all steps, but the other only some of them, but they both acquired the correct result. Thirdly, one of the most serious problems was the detection of plagiarism. In our case plagiarism is the copying of parts from submissions of the current or previous years and then passing them off as one's own. Detection of plagiarism was carried out on the basis of the good teacher's memory and collection of statistical information about task parameters, for example, chosen problem domain, labels of the nodes within the state space, etc., and peculiarities (erroneous term, incorrect usage of an algorithm, etc.), as well as registering and storing of students' submissions which were identified as sources of plagiarism. Even using such methods a significant number of course works contained plagiarism was identified every year. However, the process of plagiarism detection was very time consuming and exhausting. For plagiarism avoidance two methods were used: restriction of problem domains which students could choose, and canceling of the course work and delivery of a new individual variant if plagiarism was detected.

In order to change the situation cardinally and to eliminate the described problems, the decision to develop a template of the course work was made. The template is an MS Word document. It contains logical parts corresponding to significant solution steps of each task. In general, the following constituents form the template: explanatory text, text fields for students' text in free form, spaces for drawings and pictures, and tables for text (Figure 2).

TASK 1

Title of the problem:

<write the title of the problem which you are solving, for example, purchasing of a car>

Parameters of the task: <write the parameters of the state space which you have created>

1. number of levels within the state space = *<write your data>*
2. average branching factor = *<write your data>*

Description of the problem:

<write a brief description of the problem which you are solving>

State space:

1) Picture of the graph of the state space:

<insert a drawing of the state space>

Figure 2. Part of the template.

The template was evaluated in spring of 2007 by offering a questionnaire to students after usage of the template for the development of the course work. One hundred fifty nine questionnaires were received and processed. One hundred fifty three (96%) students pointed out that the template helped them to understand requirements for tasks performance, and six students (4%) had the opposite opinion. The same distribution of answers was received on the question, whether it was clear, what and where should be written in the template. It was offered for students to evaluate the quality of the template by using a scale, where 1 is very low quality and 5 is very high quality. One hundred and one students (64%) evaluated it with value 4, and 41 (26%) students with value 5.

On the basis of our experience and evaluation results of the template the following conclusions are made about advantages of template usage:

- from the point of view of students:
  - facilitates and reduces time of tasks completion because all necessary steps which allow to acquire solutions are clearly specified;
  - requirements for content of the course work is clearly defined;
- from the point of view of the teacher:
  - checking of submissions demands less time and efforts because course works do not contain unnecessary information;
  - all course works have the same structure and formatting that facilitates their assessment;
  - plagiarism number is reduced, because in many cases students should put in only parameters which correspond to their own individual variant.

In spite of the fact that course works created on the basis of the developed template were not checked on plagiarism by means of the special tool, we are sure that the usage of the template has decreased amount of plagiarism because it defines stricter requirements for submission formatting and content.

PLAGIARISM DETECTION TOOL FOR PROCESSING TEMPLATE-BASED DOCUMENT

The previous section demonstrates all advantages of the usage of document templates. It was concluded before that classical plagiarism detection tools are inefficient in detecting plagiarism in template-based documents due to redundant comparison procedures. Therefore we have developed the conception of a new plagiarism detection tool for processing template-based documents (Figure 3).

The main idea of the plagiarism detection tool for processing template-based documents is to make only necessary comparisons according to document content's semantic, i.e., to compare only semantically correlated parts of documents which potentially can contain identical regions.

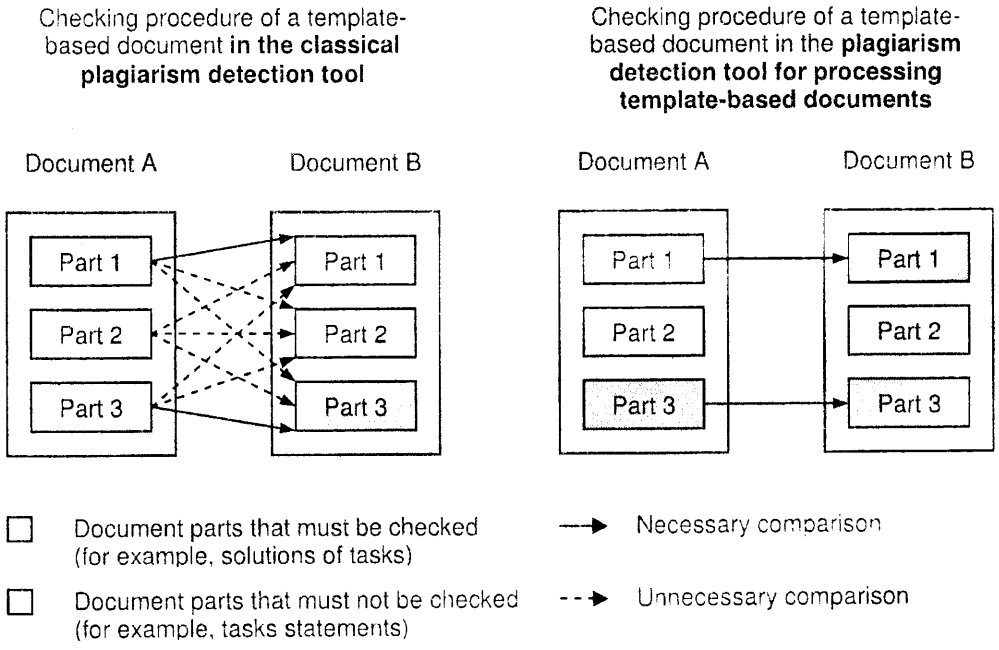

Figure 3. Comparison of documents in classical and template-based approaches to plagiarism detection.

As it was mentioned before implementation of semantic methods requires understanding of the meaning of the document, and these methods are difficult to automate. Therefore we offer to implement in the plagiarism detection tool the so called a template description function. This function would allow a user to define manually which document parts must be checked and with which parts of other document they must be compared. It helps to reduce the number of comparisons significantly and increases the speed of document processing strongly.

The checking procedure of a template-based document consists of two stages. At the first stage the document formatting is checked according to the template. All documents must have the same formatting as defined in the template; otherwise, it will not be possible to conduct mutual comparison of documents. At the second stage plagiarized regions are detected by comparing parts from one document with appropriate parts from other documents. Figure 4 demonstrates the checking procedure of a template-based document.

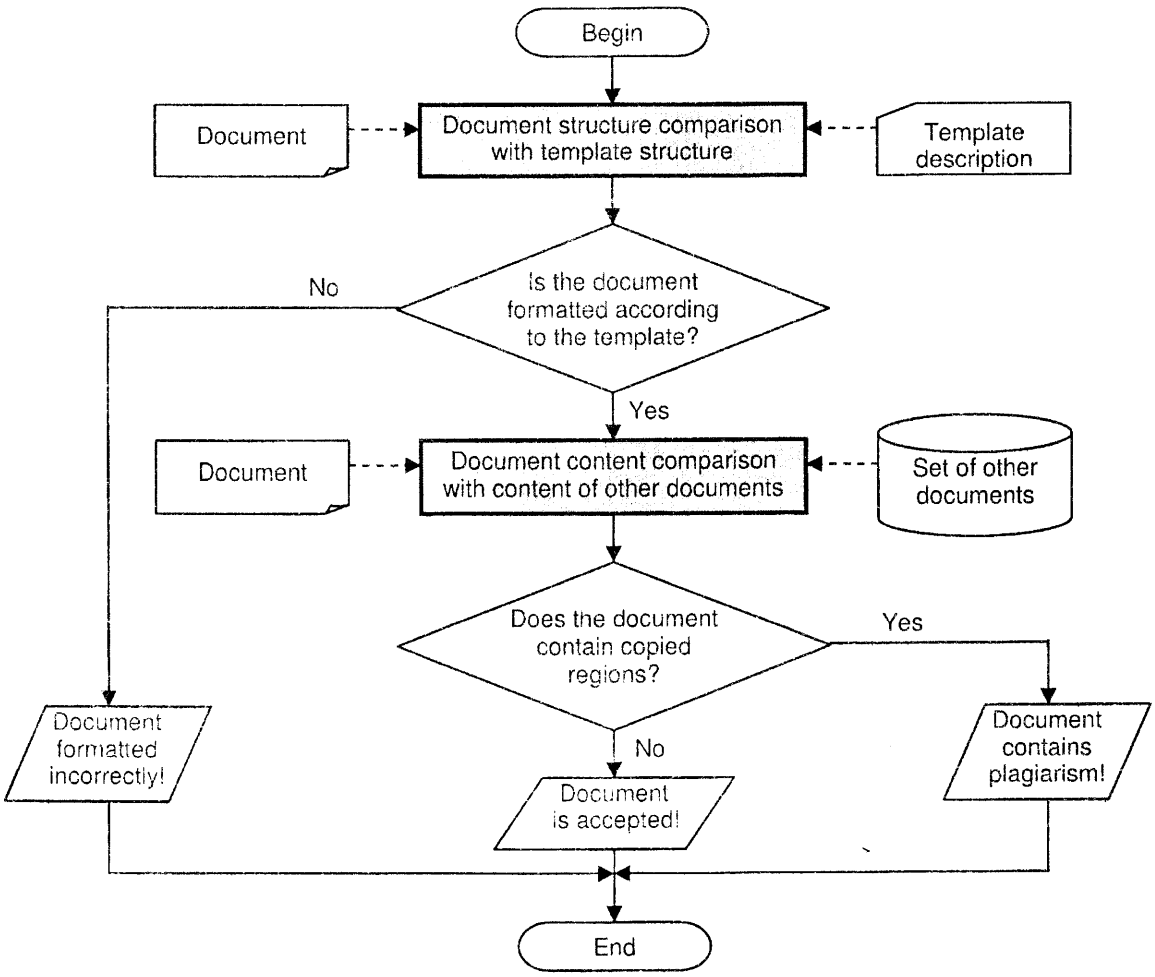

Figure 4. Checking procedure of a template-based document.

Let's describe one example to demonstrate the usage of our plagiarism detection tool for processing template-based documents. Imagine that the teacher has created the course work template for students to unify formatting of submissions. The template consists of five task statements and five blank text boxes for answers. The template was created using MS Word, where the text boxes are presented as fields with unique identifiers: Task1, Task2, Task3, Task4 and Task5.

To check students' submissions for plagiarism the teacher use the plagiarism detection tool for processing template-based documents. To use this tool properly the teacher makes some preparations. Firstly, the teacher describes the structure of the document template, i.e., defines how many fields and with what identifiers the submitted document must contain. In our example the teacher defined that each submitted document must contain five fields: Task1, Task2, Task3, Task4 and

Task5. Secondly, the teacher describes semantical relations between the fields, i.e., for each field the teacher defines fields from other documents which must be compared. In our example the tasks are absolutely uncorrelated with each other, therefore the teacher defined that Task1 from one document must be compared for similarities only with Task1 from other documents, Task2 from one document must be compared only with Task2 from other documents, etc.

After adjustment of the tool, the plagiarism detection process can be started. Each incoming document goes through two checking procedures as described before. Firstly, it is checked whether the document contains five fields with identifiers Task1, Task2, Task3, Task4 and Task5. If the document is formatted incorrectly, then it is not accepted for further checking; if the document is formatted correctly, then it goes to the similarities detection block. During the second procedure, it is checked whether the document contains similar regions with other already processed documents. In our example it is checked whether Task1 contains similar text areas with Task 1 from other documents, whether Task2 contains any similarities with Task 2 from other documents, etc. If the document contains copied regions, then it is marked as plagiarism; if the document does not contain copied regions, then it is marked as an original work. All documents, which are marked as plagiarism, afterwards can be reviewed by the teacher to make the final decision whether or not the document must be considered as plagiarism.

Taking into account the common working scheme of plagiarism detection tools (Fig.1.) and the specificity of the conception proposed in this chapter the following requirements for the plagiarism detection tool for processing template-based documents are defined:

1) Support of several user sessions running simultaneously.
2) Support of three user groups with appropriate access rights and privileges:
   - Administrator
     - management of users by adding and deleting them as well as granting privileges
   - Tutor
     - registering of new submissions
     - searching and filtering of registered submissions
     - adding of comments/keywords to submissions
     - describing the template of submissions
     - checking submissions' formatting according to the template and reviewing the results
     - checking submission's content for plagiarism and reviewing the results
   - Registering clerk
     - registering of new submissions
     - searching and filtering of registered submissions
     - checking submissions' formatting according to the template and reviewing the results
3) Authorization of users accessing the system through the usage of a user name and a password.
4) Uploading of students' submissions delivering in Word format and saving of submissions in the system's database (the system must generate a unique name for each submission in the form of NameOfTheStudent_SurnameOfTheStudent_Year).
5) Searching and filtering of submissions stored in the system's database using different parameters, at least student name and surname, date interval of submission registration, submission status (plagiarized or not plagiarized).
6) Offering of the template description function for: a) the specifying the structure of a submission template, i.e., how many parts (input fields) must be in the template and what identifiers they must have; b) the specifying semantic relations between submission's different parts (input fields), i.e., which document parts must be checked and with which parts of other documents they must be compared..
7) Displaying the submission structure and content, as well as adding comments and/or keywords to the submission.
8) Checking of submission's formatting according to the template, i.e., comparison of the submission structure with the template structure. If submission is formatted incorrectly the user should receive an error message with explanation of inconsistency. After the checking the user can accept and save the submission or deny it without registration in the system. Checking of the submission structure must be run in the interactive mode.
9) Checking of the submission's content for plagiarism by comparing the submission's content with the content of other submissions stored in system's database. If the submission contains copied regions the user should receive a list of possible sources (other submissions containing the same regions) and degree of matching with each source. After the checking the user can delete, save or print the results of plagiarism detection process. Checking of the submission content must be run in the background mode.

As it was mentioned before our plagiarism detection tool for processing template-based documents works much faster than classical plagiarism detection tools, because it does not make unnecessary comparisons of semantically unrelated parts of documents. For example, to compare two documents from our example the classical plagiarism detection tool will make 25 comparison procedures, because each of 5 parts from one document will be compared with each of 5 parts from another document. In contrast, the plagiarism detection tool for processing template-based documents will make only 5 comparison procedures, because each of 5 parts from one document will be compared with only 1 part from another document. As we see our tool will work

five times faster even processing so small document. If number of parts in the document increases, our tool will demonstrate still better results.

CONCLUSIONS

In the age of increasing usage of information technologies plagiarism has become a very topical issue and turned into a serious problem, especially for educators. Nowadays existing plagiarism detection tools provide good service in detecting matching between documents. But these tools are not very successful in detecting plagiarism in template-based documents because they make redundant comparisons of semantically unrelated parts of documents; hence, computational time is used inefficiently.

Detecting similarities in template-based documents is an actual task for educational environments, because many teachers prefer to make templates of submissions for their learning courses. Our research shows that there are many advantages in using documents' templates both for teachers and students. From one hand, the template helps teachers to unify submissions formatting and, therefore, make easy submissions checking. From the other hand, the template helps students to understand submission requirements and, therefore, facilitates submission creation.

We have introduced the conception of a new plagiarism detection tool for processing template-based documents which main idea is that a user manually defines semantical relations between document parts, i.e., for each field the teacher defines fields from other documents which must be compared. Thus the plagiarism detection tool makes only necessary comparisons of documents parts which potentially can contain identical regions. It helps to reduce the number of comparisons significantly and increases the speed of document processing strongly.

The presented plagiarism detection tool for processing template-based documents can be really useful for educational environments. Using this tool it becomes possible for the teacher quickly to find plagiarism in a set of monotype students' submissions which are created using a template.

Our future work will be concentrated on practical implementation of the plagiarism detection tool for processing template-based documents. On the basis of theoretical conception we are building a software tool and plan to test it on a set of students' submissions.

ACKNOWLEDGEMENT

The main results are outcomes of the research project ZP-2006/06 "Development of the intelligent system's prototype for plagiarism detection in students' works".

REFERENCES

Lukashenko R., Graudina V. and Grundspenkis J. 2007. Computer-Based Plagiarism Detection Methods and Tools: An Overview. Proceeding of the International Conference on Computer Systems and Technologies-CompSysTech'07, Rousse, Bulgaria, June 14-15, pp. IIIA.18-1 – IIIA.18-6.

Maurer H., Kappe F. and Zaka B. 2006. Plagiarism-A Survey. Journal of Universal Computer Sciences, vol. 12, no. 8, pp. 1050-1084.

Delvin M. 2002. Plagiarism Detection Software: How Effective is it? Assessing Learning in Australian Universities. Available at: http://www.cshe.unimelb.edu.au/assessinglearning/docs/PlagSoftware.pdf (visited 2007, April)

Lancaster T. and Culwin F. 2000. A Review of Electronic Services for Plagiarism Detection in Student Submissions. Proceedings of the 8th Annual Conference on the Teaching of Computing, Edinburgh, UK, July 23-25, 2000. Available at: http://www.ics.heacademy.ac.uk/events/presentations/317_Culwin.pdf (visited 2007, April)

Lancaster T. and Culwin F. 2005. Classifications of Plagiarism Detection Engines. ITALICS Vol. 4 (2), 2005. Available at: http://www.ics.heacademy.ac.uk/italics/Vol4-2/Plagiarism%20-%20revised%20paper.pdf (visited 2007, April)

Neill C.J. and Shanmuganthan G. A. 2004. Web-enabled Plagiarism Detection Tool. IT Professional, vol. 6, issue 5, pp. 19–23.

The University of Sydney Teaching and Learning Committee. 2003. Plagiarism Detection Software Report. Draft One. Available at: www.usyd.edu.au/su/ab/docs/2003/ABAgAug03.pdf (visited 2007, April)

Aslam J.A. and Frost M. 2003. An Information-theoretic Measure for Document Similarity. Proceedings of the 26th International ACM/SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, July 28-August 01, pp. 449–450.

Brin S., Davis J. and Garcia M.H. 1995. Copy Detection Mechanisms for Digital Documents. Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, pp. 398–409.

Lancaster T. and Culwin F. 2004. A Visual Argument for Plagiarism Detection Using Word Pairs. Proceedings of Plagiarism Prevention, Practice and Policy Conference, Newcastle, UK, June 28-30. Available at: http://www.jiscpas.ac.uk/conference2006/documents/abstracts/2004abstract15.pdf (visited 2007, April)

Tan C.L, Huang W., Sung S.Y., Yu Z. and Xu Y. 2003. Text Retrieval from Document Images Based on Word Shape Analysis. Applied Intelligence 18, pp. 257–270.