Requirements of the Plagiarism Detection Tool for Processing Template-Based Documents

Alla ANOHINA, Janis GRUNDSPENKIS
Department of System Theory and Design, Riga Technical University, Latvia

Abstract. The paper presents a detailed set of functional requirements of the plagiarism detection tool for processing template-based documents. Each function is described specifying its purpose, input data, output data and processing. The main categories of tool's users are identified. The scenario of the system's usage in an educational environment is given.

Keywords. plagiarism detection, template-based documents, requirements

Introduction

Nowadays one can consider rapid penetration of information technologies in our everyday life as well the arising of the information society. This influences also the teaching and learning processes at universities. Many different systems based on modern information technologies emerge, for example, e-learning, m-learning, intelligent tutoring systems, and many others. Besides, the Internet plays more and more important role as the information source. Moreover, reports and presentations on various topics, full solutions of tasks given in text books, answers of examination questions and even completed course papers and individual works may be found in the Internet. This is an ever growing challenge for the university staff because some part of students uses Internet offers to get needed works and submit them as their original ones. Of course, one can declare that those are weakly motivated students who do not care for their knowledge, and that the percentage of them is not very high, but for all that one should not forget it is a serious ethical problem and one of the most important educational functions of the academic staff is to enable students' fair behavior.

The experience obtained at the Faculty of Computer Science and Information Technology of Riga Technical University affirms the abovementioned declaration. The curriculum "Computer Systems" includes a number of courses such as "Database Management Systems" and "Foundations of Artificial Intelligence" where individual works are included. The teachers of these subjects must generate tasks of individual works. It is not an easy job because the number of students registered for the mentioned courses each year is around three hundreds. So, to avoid plagiarism the teachers must provide really individual tasks for each student. In fact there are not much options because it is impossible to generate several hundreds truly original tasks each year. Instead it is possible to increase the number of different tasks in two ways. First, in case if an individual work includes several separate tasks versions may be generated

randomly choosing a task from a corresponding set of tasks. In this case a probability that two or more individual works include the same tasks is very low. Of course, it is possible also to vary input data for each task. Another way to increase the number of different versions is to allow students to choose their own problem domains in which they demonstrate their solutions since each problem domain has its own semantics, input data and characteristics. Our experience to use all mentioned approaches simultaneously gives good results- it is possible to generate several hundreds different individual tasks yearly.

Unfortunately the serious drawback also has been discovered. The growing number of individual tasks causes dramatic increase of workload of teachers for checking and assessment of submitted student's works in particular if an essay-form is used. That is one of the main reasons why we propose to use a template form assignments (individual tasks). It is worth to point out that our experience confirms that the usage of essay-form assignments does not decreases the possibilities of plagiarism. On the contrary, the number of plagiarism was higher and higher each year. After introduction of the abovementioned method of variation of individual tasks the number of plagiarism starts to decrease. At the same time, sadly but the fact, that there are still tens of submitted works that contain indications of plagiarism which sources are the works of students' course-mates and works submitted in previous years found in the Internet. That is why the plagiarism detection tool is needed.

It is known that templates allow the unification of formatting rules and provide easy checking of submitted works. On the basis of our experience of the template usage within the study course "Fundamentals of Artificial Intelligence" for the third year students of the bachelor study program at the Faculty of Computer Science and Information Technology the following conclusions are made about advantages of template usage [1]:

- From the point of view of students:
  - a template facilitates and reduces time for tasks completion because all necessary steps fro the obtaining of solutions are clearly specified;
  - requirements for content of the course work are clearly defined;
- From the point of view of the teacher:
  - all course works have the same structure and formatting that facilitates their assessment;
  - assessment of the submitted students' works demands less time and efforts because course works do not contain unnecessary information;
  - number of plagiarisms is reduced, because in many cases students should put in only those parameters which correspond to their own individual version.

However, in our previous work [1] after the analysis of the already developed plagiarism detection tools such as Turnitin, Eve2, CopyCatchGold, WordCheck, Glatt, Moss, JPlag [2, 3, 4, 5, 6, 7], etc., we have found that they are inefficient for processing documents based on a template, for example, processing a set of students' individual works which are created using a template prepared by a teacher. In brief there are two main problems:

- Typically 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 parts. It distorts the overall

results of the plagiarism detection process and causes difficulties to make decisions about real amount of plagiarism in the document.

- Known plagiarism detection tools compare all parts of a document with all parts of an 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 the document processing.

Taking into account the previously described problems and our experience we have gathered requirements and developed a vision of the plagiarism detection tool for processing template-based documents.

The paper is organized as follows. In the first section the main idea of the plagiarism detection tool for processing template-based documents is presented. The second section identifies user groups of the mentioned tool, describes the process of the tool usage and specifies functional requirements in details. Conclusions and directions for future work are given at the end of the paper.

1. General Vision of the System

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 [1], i.e., to compare only semantically correlated parts of documents which potentially can contain identical regions (Figure 1).

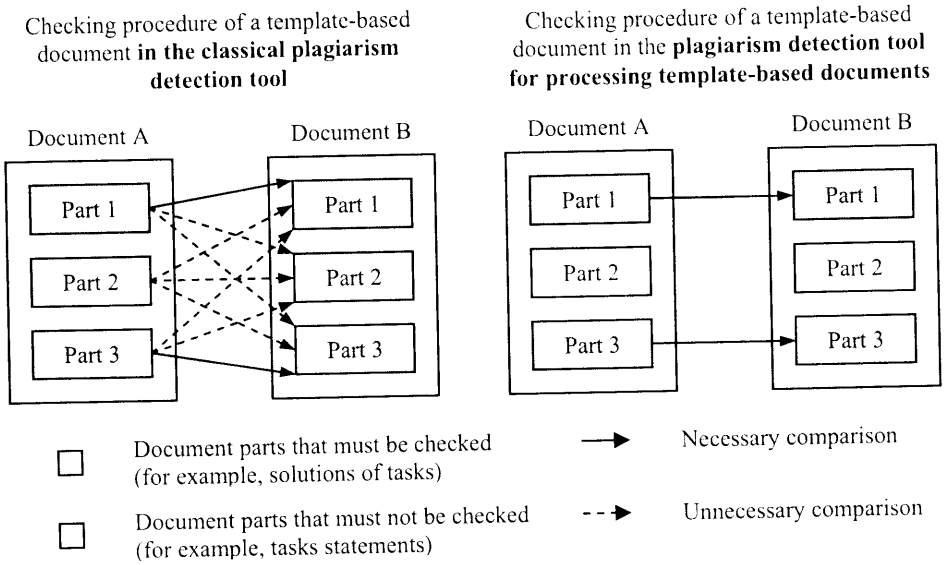

Figure 1. Comparison of documents in classical and template-based approaches to the plagiarism detection (adopted from [1])

The checking procedure of a template-based document consists of two stages. Firstly, the document formatting must be checked against the formatting of 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. Secondly, plagiarized parts of the document under checking must be detected by comparing parts from the document with corresponding parts from other documents.

It is known that document content analysis can be based either on semantic methods [8, 9] or statistical methods [9, 10]. However, the last ones are the most widely used, because they do not demand to understand the meaning of the document as it is required in semantic methods. Therefore, for the plagiarism detection process we have chosen one of the popular purely statistical methods, i.e., N-gram technique [9, 11]. Before describing the technique two definitions should be given. A *gram* is a sub-string which has a definite length. A *pattern* is a list of grams extracted from a text.

The comparison of two chunks of text includes two steps:

1. Obtaining patterns from both chunks of text.
2. Comparison of the acquired patterns.

Obtaining of a pattern is a process in which sub-strings of a definite length (grams) are extracted from the text. The sub-strings will be used for the comparison of two chunks of text and making conclusions about their similarity: texts are similar if they have similar grams. Before extraction of sub-strings the text must be modified in the following way:

- All symbols which do not belong to the alphabet (blank characters, numbers, punctuation signs, special symbols) are eliminated from the text.
- All uppercase letters are replaced by lowercase letters.

For example, in order to extract all 5-grams (sub-strings of 5 symbols) for the sentence "May 15 is my birthday!" the following steps should be performed:

1. Elimination of symbols which do not belong to the alphabet: "Mayismybirthday".
2. Replacing of letters: "mayismybirthday".
3. Extraction of the grams: "mayis", "ayism", "yismy", "ismyb", "smybi", "mybir", "ybirt", "birth", "irthd", "rthda", "thday".

When the lists of grams are acquired for both chunks of text they must be compared. As a result of this process the list with identical grams in both chunks is received. After that the similarity degree between two texts is calculated using the following equation:

$$SD = ((LP) / (LT)) * 100, \quad (1)$$

where SD is a similarity degree in percentage, LP- a number of identical grams, LT- a number of grams for the document under checking.

2. Detailed Requirements

In our previous work [1] we have defined the general requirements of the plagiarism detection tool for processing template-based documents. At the moment the full list of detailed requirements has been elaborated which we present in this paper. However our understanding of a template should be given before that.

A template is an e-document of pre-designed layout in MS Word format, which provides consistent format and content of data put into it and is based on tables and text fields with unique identifiers for input of student's data corresponding to an individual task version (Figure 2). At present only MS Word format is considered because this text processor is widely accessible for students of our university. However in the future there are plans to support other formats too, for example, OpenOffice.

Requirements gathering was made taking into account the process described in [12] and the following methods were used:

- Analysis of the existent business processes and modeling of the future business processes with the purpose to identify, which processes the system will facilitate, which changes should be made, as well as who are the potential users of the system.
- Interviewing of the system's users in order to identify how they are performing their functions now and which requirements concerning the system they have.
- Modelling of the system's functional requirements by using use cases [13].

TASK 3

Search method: *<point out the direction and the method of search>*
[ ] goal-driven search
[ ] data-driven search

[ ] breadth-first search
[ ] depth-first search

Task parameters:

a) Start state = *<write Your data>*
b) Goal state = *<write Your data >*

Implementation of the search:

Iteration	OPEN	CLOSED
0.	*<write Your data>*	*<write Your data>*
1.	*<write Your data>*	*<write Your data>*
2.	*<write Your data>*	*<write Your data>*
3.	*<write Your data>*	*<write Your data>*

Figure 2. Part of the template

2.1. System's Users

In general the system must support three groups of users with appropriate access rights and privileges:

- System administrator's responsibilities are related to the administration of the system configuration and access rights of other users. The standard functions of the user management, i.e., adding and deleting users, as well as granting

them privileges, are widely used in other software systems and therefore are not specified in this paper. Usually administrators of computer classes or departments' laboratory assistants correspond to this group.

- Tutor provides meta-data defining and checks the originality of each submitted student's work. As a rule, the role of the tutor plays professors, lecturers or assistants.
- Secretary registers submitted students' works and makes checking of their formatting against the formatting of the template. Typical examples of users of this group are a secretary of the department or of a particular professor, as well as assistants.

Authorization of users must be made through the usage of a user name and a password which must be stored in the system in the encrypted way. Multiple user sessions must be supported without the restrictions of activities of any user group.

Figure 3 displays the main functions of two user groups: a tutor and a secretary. All functions are described further in the paper.

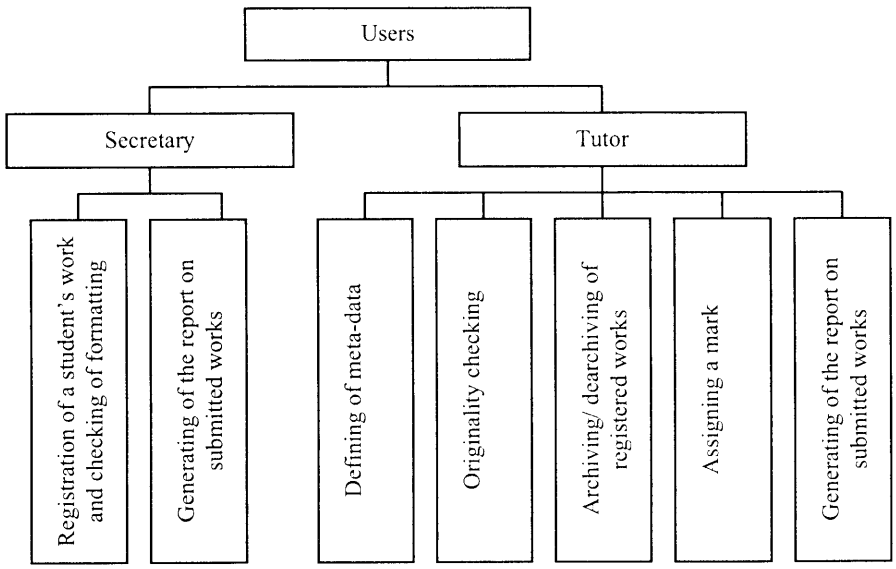

Figure 3. Main functions of the tutor and the secretary

2.2. Usage of the System

The usage of the system consists from several consecutive steps:

1. In the preparation step the teacher must describe the structure of the template by defining how many elements (text fields and tables) and with what identifiers the submitted document must contain, as well as what elements must be compared in two documents created on the basis of the same template. The description of this feature can be found in [1]. It will be elaborated after the first prototype of the system will be implemented and evaluated.

2. The secretary must register the student's submitted work in the system and must perform checking of its formatting. As a result of this process the student's work can be recognized as consistent or inconsistent with the formatting of the template. In the first case it must be excluded from the further processing. Otherwise, the subsequent steps described below must be applied to the student's work.
3. The teacher must define meta-data of the student's submitted work.
4. The teacher must perform the checking of the originality of the student's submitted work. As a result of this process the student's work can be recognized as containing or not containing plagiarized parts. In the first case the work must be excluded from the further processing. Otherwise, the last step must be applied to the student's work.
5. The teacher must assess the student's work and give it a mark.

Going through the process described above the student's submitted work can have different statuses:

- "Inconsistent formatting"- the student's submitted work has been recognized as inconsistent with the formatting of the template and therefore is excluded from the further processing.
- "Meta-data input"- the student's submitted work has been recognized as consistent with the formatting of the template and is waiting when the teacher will define its meta-data.
- "Originality checking"- the student's submitted work has been recognized as consistent with the formatting of the template, its meta-data have been defined and it is waiting when the teacher will check it for plagiarism.
- "Plagiarism" - the student's submitted work has been recognized as plagiarism.
- "Non-plagiarism"- the student's submitted work has been recognized as an original work.

2.3. Functions of the Secretary

Two main functions are defined for the secretary (Figure 3):

a) the registration of a student's work and checking of its formatting;
b) generating of the report on submitted works.

All the functions are described in the following way here and in Section 2.4: function purpose, input data specifying its necessity (mandatory or optional) and source (from the user or from the system), output data specifying its necessity (mandatory or optional) and destination (on display or within the system), as well as step-by-step processing.

Registration of a student's work and checking of its formatting

Function purpose: To register a student's submitted work in the system and to check its formatting against the formatting of the template. As a result of this process works which have inconsistent formatting are excluded from further processing.

Input:

- student's work (mandatory; from the user);
- date when the student's work was submitted (mandatory; from the user).

Output:

- summary of checking results (mandatory; on display):
  - consistency status: either inconsistent or consistent formatting;
  - identification of inconsistent parts of the work;
- student's work with the status either "inconsistent formatting" or "meta-data input" and date when it was submitted (mandatory; within the system).

Processing:

1. Choosing of the student's work by using a standard file dialog.
2. Input of the date when the student's work was submitted.
3. Checking the formatting of the submitted work against the formatting of the template:
   a) checking of the student's data within the corresponding text fields on the title-page of the student's work: first name, last name, number of the student's identity card;
   b) checking whether the student's work contains text fields or tables which are not defined within the template. At this point it is necessary to give additional explanations. As it was mentioned earlier in the paper the template consists from the tables and text fields (elements) with unique identifiers. The process of the originality checking must compare content of the elements with the identical identifiers within a pair of documents. If the student replaces some elements in his/her work with new ones or with ones taken from an other student's work, identifiers will differ from identifiers of the template. In this case it will not be possible to check the originality of their content in comparison with other documents. Therefore, it is necessary to verify that the student's work does not contain elements with different identifiers before the checking of the originality.
4. Displaying the summary of the checking results by providing information about the student's work consistency with the formatting of the template. In case of inconsistency the identification of inconsistent parts of the work must be given.
5. Saving the student's work in the system's data base:
   a) if the student's work is inconsistent with the formatting of the template, it must be stored in the system's data base by attaching it to the corresponding student with the status "inconsistent formatting";
   b) if the student's work is consistent with the formatting of the template, the system must generate a unique name for each submission in the form of NoOfStudentIdentityCard_DataOfSubmission_VersionNumber.doc, where NoOfStudentIdentityCard must be taken from the corresponding text fields on the title-page of the student's work, DataOfSubmission is the date provided in the 2nd step of this function, and VersionNumber is a version number of the work for a particular student that the system must acquire automatically from the data base, taking into account the number of the student's identity card. The student's work must be stored in the system's data base by attaching it to the corresponding student with the status "meta-data input".

Generating of the report on submitted works

Function purpose: To generate the report of students' submitted works selected by using different searching criteria.

Input:

- searching criteria (optional; from the user).

Output:

- report on students' submitted works (mandatory; on display).

Processing:

1. Searching of students' works by using the following criteria:
   a) number of the student's identity card;
   b) first name and last name;
   c) date interval of when students' works were submitted or a particular date;
   d) status of students' works;
   e) mark.

   If all criteria are not provided, then the report must include information about all students' submitted works in the current semester.
2. Displaying of the report. The report is a list of students' works which includes numbers of the student's identity cards, first names and last names, dates of work submission, marks (if the mark is not assigned to the work, the field must be empty) and status.
3. The possibilities to save the report as a Word document and to print it out must be provided, too.

2.4. Functions of the Tutor

The following main functions are defined for the tutor (Figure 3):

a) defining of meta-data;
b) originality checking;
c) archiving/ dearchiving of registered works;
d) assigning a mark;
e) generating of the report on submitted works.

Defining of meta-data

Function purpose: To specify meta-data that give additional information about the student's work and may help in the process of the originality checking.

Input:

- keywords both for the whole student's work and its particular parts (optional; from the user).

Output:

- keywords both for the whole student's work and its particular parts (optional; within the system).

Processing:

Meta-data include keywords about both the whole student's work and its particular parts. Each keyword can consist from one or more words. Their input must be organized by using an input field and a list that displays all unique keywords defined in the past for a particular part of the work. When a keyword is entered in the input field, the list must display words beginning with the same letter/letters. It will allow the user

to check, whether the keyword is already defined. If the entered keyword is new, then the possibility to add it to the list must be provided. Keywords may be undefined for a particular part of the work. If a particular part of the work is not performed by the student, then the label "Not performed" must be assigned.

Originality checking

Function purpose: To detect either the student's submitted work is an original work or it contains plagiarized parts.

Input:

- search criteria (optional; from the user);
- selected student's works or all students' works registered in the system (mandatory; from the system);
- student's work under checking (mandatory; from the system).

Output:

- checking results (mandatory; on display):
  - information about the work under checking (student's first name and last name, number of the identity card, submission date, and version number);
  - list of other students' works which parts are similar to the work under checking.
- information about plagiarism (optional; within the system):
  - name of the work which parts are similar with the work under checking;
  - parts of the work where the similarity was detected.
- status of the student's work under checking: either "plagiarism" or "non-plagiarism" (mandatory; within the system).

Processing:

1. Selection of students' works which will participate in the checking process. The following criteria can be used:
   a) number of the student's identity card;
   b) first name and last name;
   c) date interval when students' works were submitted or a particular date;
   d) status of students' works;
   e) mark.

   The selection results must be displayed as a list. The user must have the possibilities to select all works from the list or only some of them. The user must have the possibility do not make the selection. In this case the student's work will be compared with all other works registered in the system.
2. The one-to-one comparison of the documents (the submitted student's work with the selected works) will be made on the basis of the teacher's provided information about what fields must be compared in two documents created on the basis of the same template (Section 2.2). The comparison will be performed on the basis of N-gram technique described earlier in the paper (Section 1).
3. After the checking process, the system must display checking results for the user. The results must include information about the work under checking (student first name and last name, number of the identity card, submission date, and version number) and the list of other students' works which parts are similar to the work under checking. The list must contain the information

about the name of the work, work parts which are similar with the work under checking, and the similarity degree calculated in accordance with Eq. (1).

4. The possibility to compare students' submissions manually must be provided. In this case the teacher must choose one of the works from the list described in Step 3. The work under checking and the selected work must be displayed in a separate window.
5. The user can mark the students' works in the list described in Step 3 as "source of plagiarism" or "not source of plagiarism". If the work under checking is recognized as plagiarism, the following information must be stored in the system's data base: name of the work which parts are similar to the work under checking, parts where the similarity was detected, as well as the status of the work under checking must be changed to "plagiarism". Otherwise, the work will acquire the status "non-plagiarism".
6. The possibilities to print out or to save the checking results must be provided.
7. The user will not be allowed to start the checking of an other work until all works in the list described in Step 3 will not be marked as "source of plagiarism" or "not source of plagiarism".

Other functions:

1. Archiving/ dearchiving of registered works. The user must be able to select a set of registered students' works by using definite criteria (number of the student's identity card, first name and last name, date interval when students' works were submitted or a particular date, status of students' works, and mark) and to archive them. The works in the archive do not participate in the process of the originality checking. The opposite function must be used to move the works from the archive back to the active set of works.
2. Assigning a mark to the student's work. The mark can be assigned only to the work which has the status "non-plagiarism".
3. Report on submitted works. The same as the function of the secretary (Section 2.3).

3. Conclusions and Future Work

The wide expansion of the Internet is a serious challenge for the staff of educational institutions. The reason is the availability of reports and presentations on various topics, full solutions of tasks given in text books, answers of examination questions and even completed course papers and individual works within the World Wide Web that promotes growth of a number of plagiarisms among works of students. In spite of the fact that such methods as random selection of an individual task from the pool of tasks, varying of input data for each task and opportunities for students to choose their own problem domains give several hundreds different individual tasks, they do not eradicate a problem of plagiarism. In this case sources of plagiarism usually are the works of students' course-mates and works submitted in previous years found in the Internet. That is why we propose to use template-based individual tasks and the corresponding tool for the plagiarism detection.

The paper presents the vision of the plagiarism detection tool for processing template-based documents. The main idea is to compare only semantically correlated parts of documents which potentially can contain identical regions. The detailed set of

functional requirements is described specifying purpose, input data, output data and processing of each function. Three categories of tool's users are identified: a system administrator, a secretary and a tutor.

Future work is related with the design and implementation of the tool on the basis of the developed requirements specification. There are plans to evaluate the system experimentally in the study course "Fundamentals of Artificial Intelligence" after the implementation.

Acknowledgements

Authors would like to thank Dr.sc.ing. Agris Nikitenko and M.sc.ing. Romans Lukashenko for their contribution in requirements gathering and developing of the requirements specification.

References

[1] Lukashenko R., Anohina A , Grundspenkis J. A Conception of a Plagiarism Detection Tool for Processing Template-Based Documents. Annual Proceedings of Vidzeme University College, Valmiera, Latvia, 2007 (in print).

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

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

[4] Lancaster T., Culwin F. 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, pp. 54-61.

[5] Lancaster T., Culwin F. Classifications of Plagiarism Detection Engines. ITALICS, vol. 4, no. 2, 2005. Available online: http://www.ics.heacademy.ac.uk/italics/Vol4-2/Plagiarism%20-%20revised%20paper.pdf (visited April 2007).

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

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

[8] Aslam J.A., Frost M. 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, 2003, pp. 449–450.

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

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

[11] Tan C.L, Huang W., Sung S.Y., Yu Z., Xu Y. Text Retrieval from Document Images Based on Word Shape Analysis. Applied Intelligence, no. 18, 2003, pp. 257-270.

[12] Wiegers K.E. Software Requirements (2nd edition). Washington: Microsoft Press, 2003.

[13] UML Use Case Diagrams: Tips and FAQ. In: Object-Oriented Analysis and Design Course, Heinz School of Public Policy and Management, 1999. Available online: http://www.andrew.cmu.edu/course/90-754/umlucdfaq.html (visited January 2008).