Information Source Detection and Anti-Plagiarism System

By:

Lenard Osmani

June 2012

To My Parents

INFORMATION SOURCE DETECTION AND APTI-PLAGIARISM SYSTEM

By

Lenard Osmani

June 2012

Chairman: Igli Hakrama

Faculty: Faculty of Architecture and Engineering

Abstract: The availability of textual material in electronic format has made plagiarism easier than ever. Copying and pasting of paragraphs or even entire essays now can be performed with just a few mouse clicks. The strategies discussed here can be used to combat what some believe is an increasing amount of plagiarism on research papers and other student writing. By employing these strategies, you can help encourage students to value the assignment and to do their own work. But there is much to be done and this is the long way around.

There are also many ways to plagiarism prevention, and there is an ever growing need for protection and verification of copyright. Before checking the document for plagiarism, reviewing algorithms and approaches for searching plagiarism, we must know and understand what constitutes the plagiarism. To cover that, this paper is divided in two parts:

The first part explains thoroughly what plagiarism is, why people do it and how to educate students not to do it.

And the second part, the paper describes the most common plagiarism detection systems, methods used in those systems, makes comparisons to point out their pros and cons and finally explains how this knowledge is used to build a real working (though It’s just a simple prototype for now) anti plagiarism system.

Keywords: –algorithms for searching plagiarism, plagiarism, plagiarism detection systems, programs for plagiarism detection.

SISTEM PËR DETEKTIMIN E BURIMIT TË INFORMACIONIT DHE ANTI-PLAGJATURËS

Autor

Lenard Osmani

Qershor 2012

Chairman: Igli Hakrama

Fakulteti: Fakulteti i Arkitekturës dhe Inxhinierisë

Abstrakti: Disponueshmëria e materialeve tekstuale në format elektronik, ka bërë plagjiaturën më lehtë se kurrë. Kopjimi i paragrafëve apo edhe teksteve te plota tani mund të kryhet vetëm me disa klikime. Strategjitë e diskutuara këtu mund të përdoren për të luftuar rritjen e vazhdueshme të plagjiaturës në kërkimet shkencore dhe shkrimit të tjera të studentëve. Duke përdorur këto strategji, ju mund të inkurajoni studentët që të vlerësojnë detyrën dhe të bërë punën e tyre origjinale.

Por ka shumë për të bërë dhe kjo është rruga e gjatë rreth ketij problemi. Ka edhe shumë mënyra për dallimin dhe ndalimin plagjiaturës, dhe ka një nevojë gjithnjë në rritje për mbrojtjen dhe verifikimin e të drejtave të autorit. Para se të kontrollohet dokumenti për plagjiaturë, duke përdorur algoritme të ndryshme për kërkimin e plagjiaturës, ne duhet të njohin dhe të kuptojnë se çfarë përbën plagjiatura.

Për të mbuluar këtë, ky liber është e ndarë në dy pjesë: Pjesa e parë shpjegon hollësisht se çfarë është plagjiatura, pse njerëzit e bëjnë atë dhe se si të edukojmë studentët për ta evituar atë.

Dhe pjesa e dytë: përshkruan menyrat më të zakonshme të sistemeve të zbulimit të plagjiaturës, metodat e përdorura në këto sisteme, bën krahasime për të vënë në dukje anët pozitive dhe negative te secilës dhe më në fund shpjegon se si kjo dije është përdorur për të ndërtuar një pune reale (edhe pse është vetëm një prototip i thjeshtë për tani), sistem anti plagjiaturë.

Fjalë Kyçe: algoritme për gjetjen e plagjiaturës, plagjiaturë, sistemet e zbulimit të plagjiaturës.

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation and deepest gratitude to his supervisor, M.Sc. Igli HAKRAMA for his kind supervision, guidance, suggestions, encouragement and advices throughout the course of this study. Thanks and sincere appreciations also go to his co-supervisor, M.Sc. Ibrahim MESECAN and M.Sc. Elton HOXHA for their guidance and constructive comments towards this project.

The author would also like to express his gratitude to all colleagues, friends and other professors for their assistance and help throughout the period of completing this project. While those are not named here, their contribution has been important as well.

- Lenard OSMANI

LIST OF PICTURES

TABLE OF CONTENTS

INTRODUCTION

According to the Collins Dictionary of the English Language, plagiarism is ‘the act of plagiarizing’, which means ‘to appropriate (ideas, passages, etc) from (another work or author)’ [7].

The verb “plagiarize” is defined in the Shorter Oxford as follows: ‘Take and use as one’s own (the thoughts, writings, inventions, etc., of another person); copy (literary work, ideas, etc.) improperly or without acknowledgement; pass off the thoughts, work, etc. of (another person) as one’s own’ [3].

However the difference between plagiarism and research is really very thin.
After all, advanced research is only possible by using the already existing information.

The core business of the knowledge industry is handling information and ideas from different sources, so there is inevitably great scope for plagiarism within the academic world. Here plagiarism occurs in a variety of settings, including collaboration or cooperation between students working together, unattributed use of other people’s writings by undergraduates, copying of graduate students’ work by supervisors or other members of academic staff and taking credit in research grant applications for work done by someone else.

Plagiarism is the old problem in the highest education that was aggravated with the advent of the Internet [11].

However, it should be noted that when the student enters the university, he might be not informed about plagiarism or how to overcome it. Therefore, plagiarists share on types that define seriousness of their actions.

Three types of plagiarists [8] can be identified:

- Accidental: lack of understanding, the student was unaware that it was wrong thus demonstrates poor academic practice;

- Opportunistic: aware of this being ‘wrong’ but does so due to some source of pressure or in the belief that it will result in higher marks;

- Committed: intentional (pre-meditated) cheating via misrepresentation.

Researches show that the majority of plagiarism carried out by students who don’t understand the academic requirements; therefore the majority of students are accidental plagiarists.

PART I: Understanding plagiarism & Strategies of prevention

CHAPTER 2

LITERATURE REVIEW & CONTEXT

2.1 - Understand why people cheat

By understanding some of the reasons different people are tempted to cheat on papers, you can take steps to prevent cheating by attacking the causes. Some of the major reasons include these:

- Many students simply do not know what plagiarism is. Their awareness, if any, often derives from urban legends and myths ("Everything on the Internet is public domain and can be copied without citation").

- Many other students know what plagiarism is, but don’t consider it wrong. The belief that "information wants to be free," and the idea that copying from sources with a few words of one’s own is merely "patch writing," a normal way to write, support these students in their beliefs. So the plagiarizer you catch might not be the defiant, lazy cheater you assume, but a practical, "community of words" compiler of essays using fellow writers’ verbal structures.

- Students are natural economizers. Many students are interested in the shortest route possible through a course. That’s why they ask questions such as, "Will this be on the test?" Copying a paper sometimes looks at the shortcut through an assignment, especially when the student feels overloaded with work already. To combat this cause, assign your paper to be due well before the end-of-term pressures. Remind students that the purpose of the course is to learn and develop skills and not just "get through." Writing a research paper helps to develop the skills of researching (hunting for something in the information universe), problem solving (the principal work of most people), writing (language is the most powerful weapon on earth), perseverance, and commitment. It follows that the more students learn and develop their skills, the more effective they will be in their future lives.

- Students are faced with too many choices, so they put off low priorities. With so many things to do (both of academic and recreational nature), many students put off assignments that do not interest them. A remedy here would be to customize the research topic to include something of real interest to the students or to offer topics with high intrinsic interest to them.

- Many students have poor time management and planning skills. Some students are just procrastinators, while others do not understand the hours required to develop a good research paper, and they run out of time as the due date looms. Thus, they are most tempted to copy a paper when time is short and they have not yet started the assignment. If you structure your research assignment so that intermediate parts of it (topic, early research, prospectus, outline, draft, bibliography, final draft) are due at regular intervals, students will be less likely to get in a time-pressure panic and look for an expedient shortcut.

- Some students fear that their writing ability is inadequate. Fear of a bad grade and inability to perform cause some students to look for a superior product. This is sometimes called "cheat to compete." Sadly, these students are among those least able to judge a good paper and are often likely to turn in a very poor copied one. Some help for these students may come from demonstrating how poor many of the online papers are and by emphasizing the value of the learning process (more on this below). Reassuring students of the help available to them (your personal attention, a writing center, teaching assistants, online writing lab sites, etc.) may give them the courage to persevere.

- A few students like the thrill of rule breaking. The more angrily you condemn plagiarism, the more they can hardly wait to do it. An approach that may have some effect is to present the assignment and the proper citation of sources in a positive light (more below).

2.2 - Educate yourself about plagiarism.

Plagiarism on research papers takes many forms. Some of the most common include these:

- Downloading a free research paper. Many of these papers have been written and shared by other students. Since paper swappers are often not among the best students, free papers are often of poor quality, in both mechanics and content. Some of the papers are surprisingly old (with citations being no more recent than the seventies).

- Buying a paper from a commercial paper mill. These papers can be good--and sometimes they are too good. If you have given students an in-class writing assignment, you can compare the quality and be quite enlightened. Moreover, mills often sell both custom and stock papers, with custom papers becoming stock papers very quickly. If you visit some of the mill sites, you might just find the same paper available for sale by searching by title or subject.

- Copying an article from the Web or an online or electronic database. Only some of these articles will have the quantity and type of citations that academic research papers are expected to have. If you receive a well-written, highly informed essay without a single citation (or with just a few), it may have been copied wholesale from an electronic source.

- Copying a paper from a local source. Papers may be copied from students who have taken your course previously, from fraternity files, or from other paper-sharing sources near campus. If you keep copies of previous papers turned in to you, they can be a source of detection of this particular practice.

- Cutting and pasting to create a paper from several sources. These "assembly-kit" papers are often betrayed by wide variations in tone, diction, and citation style. The introduction and conclusion are often student-written and therefore noticeably different from and weaker than the often glowing middle.

- Quoting less than all the words copied. This practice includes premature end quotation marks or missing quotation marks. A common type of plagiarism occurs when a student quotes a sentence or two, places the end quotation mark and the citation, and then continues copying from the source. Or the student may copy from the source verbatim without any quotation marks at all, but adding a citation, implying that the information is the student's summary of the source. Checking the citation will expose this practice.

- Faking a citation. In lieu of real research, some students will make up quotations and supply fake citations. The fake citation can be either completely fabricated (The

American Journal of Asymmetric Induction Studies), or it can reference a real source (book, journal, or Web site) which contains no such article or words that have supposedly been used. You can discover this practice by randomly checking citations. If you require several Web or other electronic sources for the paper, these can be checked quickly.

Visiting some of the sites that give away or sell research papers can be an informative experience. If you have Web projection capability, you might do this visiting in class and show the students (1) that you know about these sites and (2) that the papers are often well below your expectations for quality, timeliness, and research. There is a list of many of these Internet paper mills here. There are some good discussion points at "Cheating 101: The Danger of Using an Internet Paper Mill" from Adultlearn.com.

2.3 - Educate students and others around you about plagiarism.

Do not assume that people around you know what plagiarism is, even if they nod their heads when you ask them. Provide an explicit definition for them. For example, "Plagiarism is using another person's words or ideas without giving credit to the other person. When you use someone else's words, you must put quotation marks around them and give the writer or speaker credit by revealing the source in a citation.

Even if you revise or paraphrase the words of someone else or just use their ideas, you still must give the author credit in a note. Not giving due credit to the creator of an idea or writing is very much like lying."

In addition to a definition, though, you should discuss with your students the difference between appropriate, referenced use of ideas or quotations and inappropriate use. You might show them an example of a permissible paraphrase (with its citation) and an impermissible paraphrase (containing some paraphrasing and some copying), and discuss the difference.

Discuss also quoting a passage and using quotation marks and a citation as opposed to quoting a passage with neither (in other words, merely copying without attribution).

Such a discussion should educate those who truly do not understand citation issues ("But I put it in my own words, so I didn’t think I had to cite it") and it will also warn the truly dishonest that you are watching. Wholesale copying is obviously intentional, but a paper with occasional copy and paste sentences or poorly paraphrased material might be the result of ignorance.

Discussing with students why plagiarism is wrong may be helpful also. Clarifying for them that plagiarism is a combination of stealing (another’s words) and lying (claiming implicitly that the words are the student’s own) should be mentioned at some point, but should not be the whole emphasis or you risk setting up a challenge for the rebels (those who like to break the rules just for fun).

Many statements on plagiarism also remind students that such cheating shows contempt for the professor, other students, and the entire academic enterprise. Plagiarizers by their actions declare that they are not at the university to gain an education, but only to pretend to do so, and that they therefore intend to gain by fraud the credentials (the degree) of an educated person.

Perhaps the most effective discussion will ask the students to think about who is really being cheated when someone plagiarizes.

Copying papers or even parts of papers short circuits a number of learning experiences and opportunities for the development of skills: actually doing the work of the research paper rather than counterfeiting it gives the student not only knowledge of the subject and insights into the world of information and controversy, but improves research skills, thinking and analyzing, organizing, writing, planning and time management, and even meticulousness (those picky citation styles actually help improve one's attention to detail).

All this is missed when the paper is faked, and it is these missed skills which will be of high value in the working world. A degree will help students get a first job, but performance--using the skills developed by doing just such assignments as research papers--will be required for promotion.

2.4 - Discuss the benefits of citing sources.

Many students do not seem to realize that whenever they cite a source, they are strengthening their writing. Citing a source, whether paraphrased or quoted, reveals that they have performed research work and synthesized the findings into their own argument. Using sources shows that the student in engaged in "the great conversation," the world of ideas, and that the student is aware of other thinkers' positions on the topic.

By quoting (and citing) writers who support the student's position, the student adds strength to the position. By responding reasonably to those who oppose the position, the student shows that there are valid counter arguments.

In a nutshell, citing helps make the essay stronger and sounder and will probably result in a better grade. Most college graduates will become knowledge workers themselves, earning at least part of their living creating information products. They therefore have an interest in maintaining a respect for intellectual property and the proper attribution of ideas and words.

2.5 - Make the penalties clear.

If an institutional policy exists, quote it in your syllabus. If you have your own policy, specify the penalties involved. For example, "Cheating on a paper will result in an F on that paper with no possibility of a makeup. A second act of cheating will result in an F in the course regardless of the student's grade otherwise." If you teach at a university where the penalty for plagiarism is dismissal from the university or being reported to the Academic Dean or Dean of Students, you should make that clear as well.

Even the penalties can be presented in a positive light. Penalties exist to reassure honest students that their efforts are respected and valued, so much so that those who would escape the work by fakery will be punished substantially.

Note: There are always a few students who will be caught plagiarizing and then claim that no one cared or told them. When you point to the section in your syllabus, they will say, "I thought it was a generic syllabus so I didn’t read it." The better idea, then, is to read the appropriate places from the syllabus.

PART II: Strategies of detection, Uses and Examples

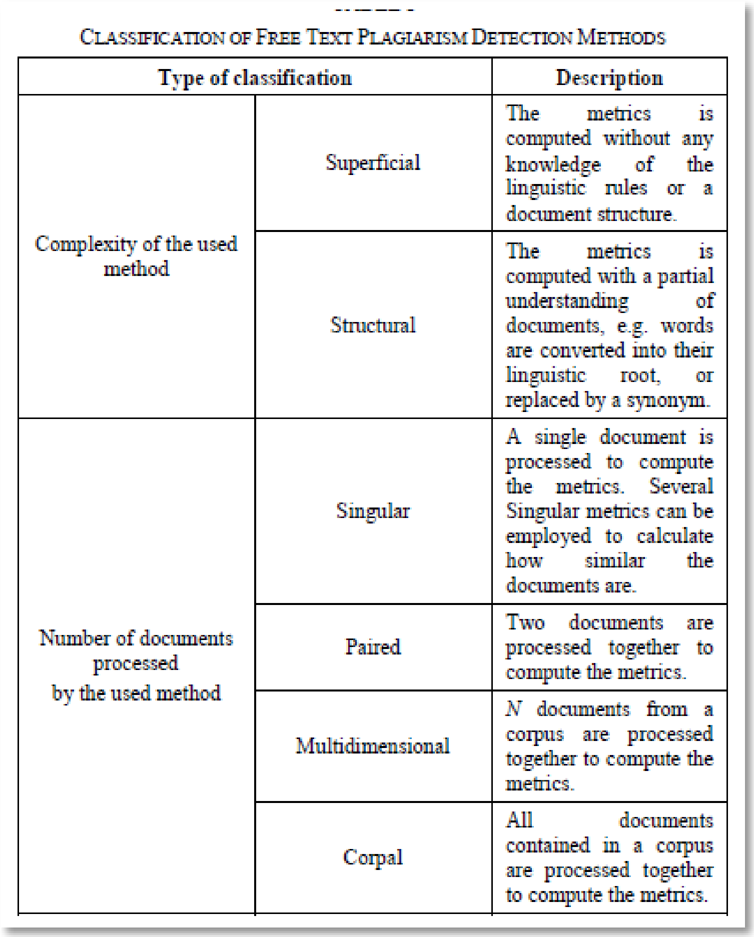
CHAPTER 3

METHODOLOGY

3.1 - Plagiarism detection methods

People can easily search for the required documents and make their copy instead of writing the documents themselves. In addition, the problem is also supported by many servers, which offer a wide range of various topics. Document protection techniques, which disable copy-paste operations and printing, are insufficient. A large database of existing documents is a better solution. The main idea of this protection rests in psychology because every plagiarized document can be easily identified when compared to the database. Most of the plagiarists only copy a part of a document and do not try to hide this activity. While the consistent plagiarists copy some parts of sentences and sometimes exchange several words to cause confusion. This type of plagiarism is difficult to determine [4].

The most general classification of copy detection methods is to free text or source code. The classification in Table 1 is intended just for free text plagiarism detection methods [4]. To check for plagiarism, every document



CLASSIFICATION OF FREE TEXT PLAGIARISM DETECTION METHODS

Type of classification		Description
Complexity of the used method	Superficial	The metrics is computed without any knowledge of the linguistic rules or a document structure.
	Structural	The metrics is computed with a partial understanding of documents, e.g. words are converted into their linguistic root, or replaced by a synonym.
Number of documents processed by the used method	Singular	A single document is processed to compute the metrics. Several Singular metrics can be employed to calculate how similar the documents are.
	Paired	Two documents are processed together to compute the metrics.
	Multidimensional	*N* documents from a corpus are processed together to compute the metrics.
	Corpal	All documents contained in a corpus are processed together to compute the metrics.

must be compared with any other possible documents to analyze the whole corpus. Therefore, these methods are suitable for seeking some possibly plagiarized documents, which are related to the concrete tested document. The older systems, such as COPS or SCAM working on the term frequency, are purely Superficial. The current systems, which employ N-grams, are also rather Superficial than Structural because too much time is spent in analysis of sentences whose grammar includes many linguistic rules. Fortunately, some modern approaches from the other fields of nature language processing give us new possibilities of improving the current plagiarism detection methods [4].

3.1.1 - The **COP (Copy Protection System)** is a prototype of copy detection system developed at Stanford. The system sketches the common approach to plagiarism detection based on unit chunk hashing. A chunk is a sequence of consecutive units; a document may be divided into chunks in a number of ways, as chunks are allowed overlap or not cover the document entirely. A method of selecting chunks from a document is called a chunking strategy.

A system following the COPS methodology consists of two main functions. One which obtains chunks from a document via a selected chunking strategy and stores hashes of these chunks into a hash table. The second function is a function that realizes the violation test [14].

3.1.2 - SCAM (Stanford Copy Analysis Mechanism) is a plagiarism detection system developed at Stanford. Unlike COPS, it operates by assuming vector space model for the registered documents. The difference to other Information Retrieval (IR) systems is in using a new similarity measure. This measure was developed to more accurately characterize copy overlap, while traditional IR systems look for semantic similarity [14].

The SCAM system, as well as COPS, is classified as paired and superficial system. It is tuned to discover small overlaps, which results in many false positives when word distributions are similar but the texts are still different [4, 15].

3.1.3 - MOSS (Measure of Software Similarity) was developed at UC Berkeley in 1994. It is a free available plagiarism detection system for academic usage only. MOSS supports a lot of different programming languages and two platforms, UNIX and Windows. As the name suggests, its primary purpose is to detect programming assignment plagiarism and is used mostly by programming lecturers from computer science and engineering departments, although it also supports other text input types apart from code. Its aim is to detect the standard attempt at cheating, which consists of changing variable names, I/O prompts, statement spacing and comments. However, even this ’dumb’ attempt is enough to fool a simple file diff, rendering a careful manual comparison necessary. MOSS overcomes this problem by offering a script which, whenever run, emails a selected batch of programs to a Berkeley server for analysis. Response is usually obtained within the same day and consists of a set of html documents comprising a report. The report highlights pairs of programs that exhibit suspiciously high mutual similarity [5, 14].

3.1.4 - YAP (Yet Another Plague) – token-based system that treats programs as a sequence of strings. The last version of YAP (YAP3) introduces a totally novel algorithm to face the presence of block-moves in programs. Namely: the Running- Karp-Rabin Greedy-String-Tiling algorithm.

Its aim is to find a maximal set of common contiguous substrings as long as possible, each of which does not cover a token already used in some other substrings [9, 14].

3.1.5 - MDR (Match Detect Retrieval) is a prototype of a system capable of detecting overlapping documents. The approach used in Match Detect Reveal system avoids using a hash function due to concern about hash collisions. The basic matching components uses string-matching algorithm based on suffix trees to identify the overlap. The algorithm used for building the suffix tree from the query document is a modification of Kekkonen's algorithm. As such, this system is only capable of locating exact copies of document parts. Once the suffix tree is built, all registered documents are compared against it [14, 15].

3.1.6 - SID (Software Integrity Diagnosis, or alternatively Share Information Distance) is a system developed at University of California, Santa Barbara. The authors noted that plagiarism detection systems like MOSS and the YAP proceed by tokenizing the input sequence and then comparing the token sequences. A basic problem underlying the second phase is how to measure similarity of a pair of token sequences. If the metric is inappropriate, plagiarism may go unnoticed. If it is well-defined and not universal, it can always be cheated. For example, the MOSS designers refuse to publish details of their algorithm openly on the website, fearing the cheaters would quickly learn to beat the system. Such an approach to security through withholding static information is not a good design choice. Authors of SID therefore take a different approach, where they consider the sequence similarity from an information theoretic perspective. The metric that measures the amount of information between two sequences (not necessarily program token sequences – many applications are imaginable, including DNA sequences or text documents) is based on Kolmogorov complexity and is universal. The universality guarantees that if there is similarity under any computable similarity metric, this metric will detect it [14].

3.1.7 - CHECK is another plagiarism detection system that uses document structure to build a hierarchal representation of the document. Each document is viewed at multiple abstraction levels, which include the document itself, its section, subsection, sub-subsections and finally paragraphs. For each level, the set of relevant keywords is extracted. Keyword extraction uses keywords frequency as well as italics and boldface formatting information to assign weights to keywords. At query time, the nodes of the query abstraction and that of the referential document are traversed, starting with the root node. Similarity is computed as cosine measure of the two node's keyword weight vectors. If the similarity exceeds a given threshold, the two node's children are processed recursively. The purpose of this step is to obtain pairs of document segments (represented by the lowest level of abstraction, i.e. paragraphs) that are similar to each other. The final step is to analyze these similar pairs of paragraphs sentence-by-sentence and report detected copies [14].

3.2 - Plagiarism Detection Algorithms

One of the plagiarism detection algorithms is the Heskel's algorithm which is based on the sharing string on k-grams that is k-length substrings, and search for matches, focusing already on them. Nevertheless, this algorithm has a principal lack. In the big programs is a very small number of unique K-grams. Therefore many coincidence that don't contain such K-grams, will be ignored [5, 10].

Another algorithm uses the method of local alignment of strings which has been developed for determination the similarity of strings of DNA (deoxyribonucleic acid). To use this method, two programs should be represented as a string of characters. Alignment of strings is obtained by inserting spaces in the strings so that their length became equal. It should be noted that there are a large number of different alignments of two strings [21].

It is necessary to consider a heuristic algorithm of greedy string tiling. It receives the input two strings of characters, and the output set of their common non-overlapping substrings, which is close to optimal. Substring is appearing in this set, called tile. There are quite a large number of optimizations of this algorithm which considerably increase its high-speed performance. There is also a more radical improvement using the algorithm of Karp-Rabin of substrings search in the string. The main advantage of the algorithm can assume that the rearrangement of the large part of the code does not affect efficiency of the algorithm [10, 13, 19].

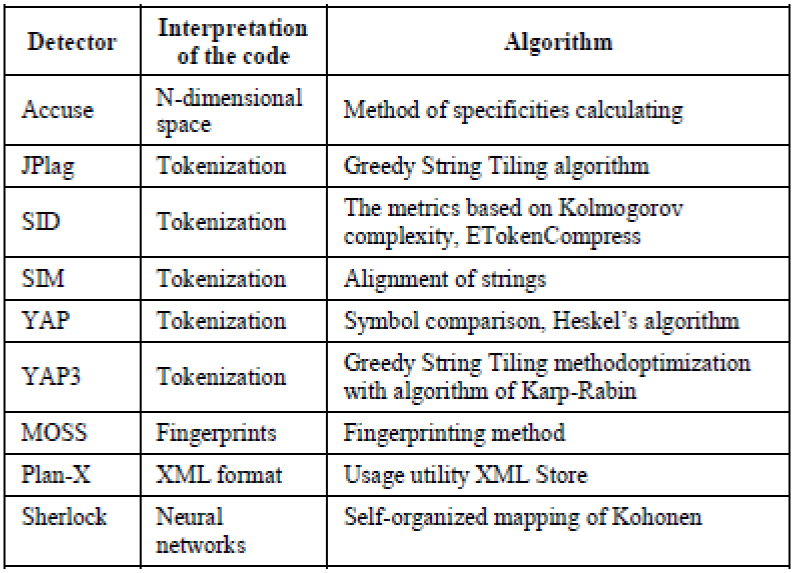
One more interesting algorithm is based on Kolmogorov complexity. The basis of this algorithm is a function of distance which is based on Kolmogorov complexity. The more close function of distance of two programs to zero, the more shared information these programs contain. As Kolmogorov complexity is not computable, the heuristic approach based on the use of compression algorithm is employed [21].

In the fingerprinting method, tokenization programs are represented as sets of prints so that similar sets of similar programs overlap. This method allows user to implement an effective search for large databases [21].

There are few algorithms that use the interpretations as a tree or a graph. Only two of these algorithms can be performed at reasonable times. Therefore, they are rarely used in practice. For plagiarism detection, a method of neural networks can also be used. Plagiarism detection can be compared to the classification task in which a set of programs can be divided into classes, in each of which there will be only copied programs.

Neural networks can be represented as the black box whose input data is the known information, the output data - the information that you would like to know. For example, the input data can be the set of programs, and the output data - the inference about plagiarism presence. This method, for example, uses the detector

Systems and Algorithms



Detector	Interpretation of the code	Algorithm
Accuse	N-dimensional space	Method of specificities calculating
JPlag	Tokenization	Greedy String Tiling algorithm
SID	Tokenization	The metrics based on Kolmogorov complexity, ETokenCompress
SIM	Tokenization	Alignment of strings
YAP	Tokenization	Symbol comparison, Heskel's algorithm
YAP3	Tokenization	Greedy String Tiling methodoptimization with algorithm of Karp-Rabin
MOSS	Fingerprints	Fingerprinting method
Plan-X	XML format	Usage utility XML Store
Sherlock	Neural networks	Self-organized mapping of Kohonen

Sherlock [22]. The summarized information is given in Table II where it can be seen what interpretation of the code and what algorithm for plagiarism detection are used by each of detectors.

3.3 - Existing Plagiarism Detection Software/Systems (In English language)

3.3.1 - Turnitin [17] is the most popular service of plagiarism detection. It was developed by group iParadigms for teachers and educational institutions and was formerly known as Plagiarism.org. The service works on a commercial basis and requires pre-registration. Professors and teacher’s present student's works on site and in a day or two receive the results. The system compares these materials to the indexed Web content, large databases containing texts from so-called “collections of essays” (they are sold in Internet for usage as school or university term papers), as well as previously reported materials [5, 23].

3.3.2 - WordCHECK [12] exposes more students copying from each other, than borrowing of external materials. To use this application the teacher downloads all documents in the internal archive where they are compared for detection of copying within educational group. Comparing is based on the profiles of keywords (a sort of linguistic equivalent of the fingerprint) and comparison of phrases [5, 23].

Although the system, strictly speaking, isn't calculated for plagiarism detection, it will be able to do it if you include in the internal archive texts from «collections of essays» and other similar materials. Unfortunately, according to the results of the tests of this tool performed in 2001 by request of committee JISC, its functional capabilities were recognized as unsatisfactory.

3.3.3 - Program EVE2 [6] — commercial application that when installed on the PC finds out whether the student has not copied material from the Internet. For every work, application generates the report containing instructions of percentage of loans, list URL and the annotated copy of the work in which the copied fragments are selected by red color. It is possible to use several file formats, including plain text and Microsoft Word documents, but the annotated copies are created only for plain text [5, 23].

In essence, this tool provides the interface to the search engine in the Internet, but such simplicity doesn't restrict its efficiency. The unique lack of EVE2, noted in report for the JISC in 2001,is that search is fulfilled only for Web-content in HTML format, but the most part of a material in the World Wide Web is stored in other formats.

3.3.4 - WCopyFind [16] — the free tool for detection of the facts of writing off by the students, developed by Professor Lu Bloomfield at the University of Virginia. The documents are compared with each other and, at will, it is possible to separate archive of files (which the professor, probably, collected several years) to compare sentences.

CHAPTER 4

SIMPLE ANTI-PLAGIARISM SYSTEM

4.1 Features of the system

The name explains a lot about this project, that due to the time and short term purpose (for now) I only built a prototype application.

The algorithm is a combination of the WordCHECK and Double Content Finder algorithms which were explained above.

It is a small system where you can upload your documents and then check the report about the originality. It has a database which runs over MySQL for now, composed of a few simple tables. In these tables, are stored the full documents, and after some complex data mining operations, it is divided into chunks. The chunks can be pages, paragraphs and subparagraphs and sentences.

Depending on the desired kind of result or check, we can choose which one of them will be the basis of the algorithm. It can compare by paragraphs, by sentences or even bigger blocks of text.

After this classification, a fingerprint of every chunk is derived. This is a kind of hash, that serves to give us more accurate results, depending on the phrase composition, word used in the phrase and the density of those word.

Thank to this fingerprint, our results are very accurate, and it is able to overdue possible changes that the cheater can have done to the copied parts.

Another important feature gained by using fingerprints, is that the system can be used in multiple languages. The files can be stored in the same table in different languages, but the fingerprint will only match results that have the same language as the one we are processing.

The whole project is processed using PHP programming language which has some advanced data handling and manipulation functions for strings or texts.

The check and clustering is done on upload, and it is called every time we want, giving us the ability to have real time results and reports.

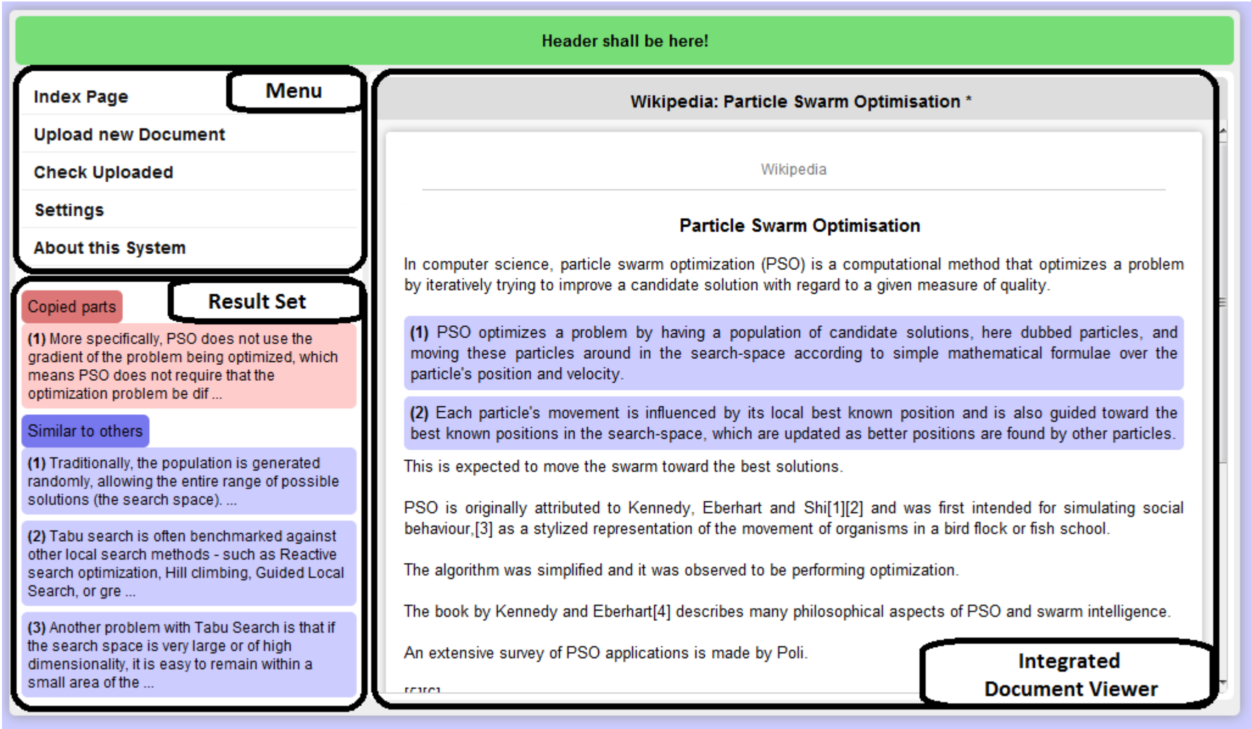
With a simple pluggin, the content is stored as a file, and can be downloaded if wanted. The same is for the report.

The system highlights the code which is copied with red background and also gives numbers to those paragraphs.

It also gives some more results, of similar paragraphs. This is done to get the closest other paper to this one, and also to detect very well hidden cases of plagiarism.

The interface is fey simple, and is shown in the screenshots below:

Pic.3: Main interface when checking a paper.



(1) More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be dif ...

(1) Traditionally, the population is generated randomly, allowing the entire range of possible solutions (the search space). ...

(2) Tabu search is often benchmarked against other local search methods - such as Reactive search optimization, Hill climbing, Guided Local Search, or gre ...

(3) Another problem with Tabu Search is that if the search space is very large or of high dimensionality, it is easy to remain within a small area of the ...

Wikipedia: Particle Swarm Optimisation *

Wikipedia

Particle Swarm Optimisation

In computer science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

(1) PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity.

(2) Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles.

This is expected to move the swarm toward the best solutions.

PSO is originally attributed to Kennedy, Eberhart and Shi[1][2] and was first intended for simulating social behaviour,[3] as a stylized representation of the movement of organisms in a bird flock or fish school.

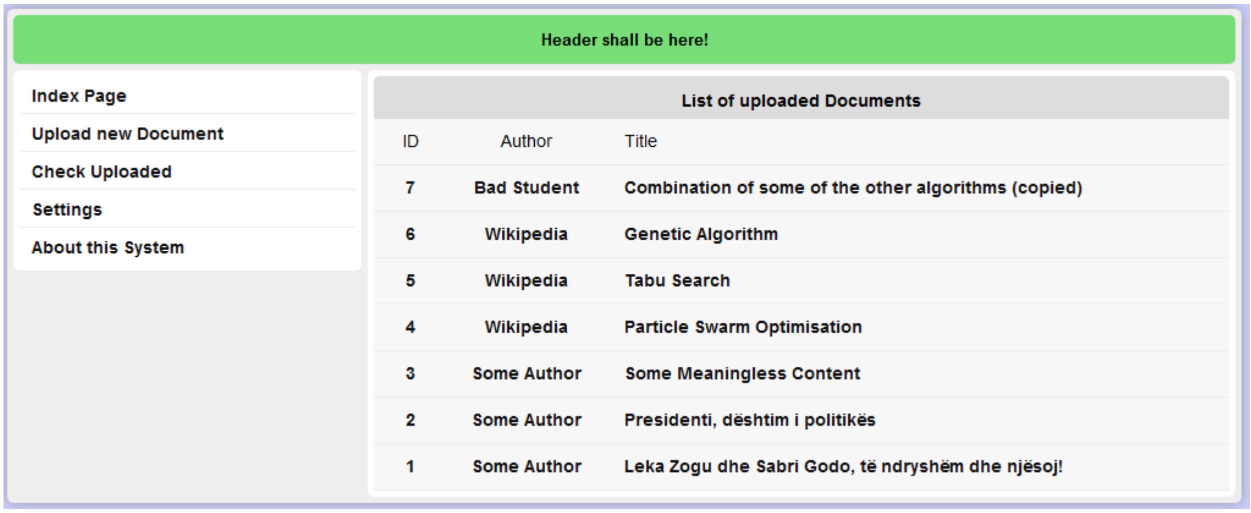
The algorithm was simplified and it was observed to be performing optimization.

The book by Kennedy and Eberhart[4] describes many philosophical aspects of PSO and swarm intelligence.

An extensive survey of PSO applications is made by Poli.

[5][6]

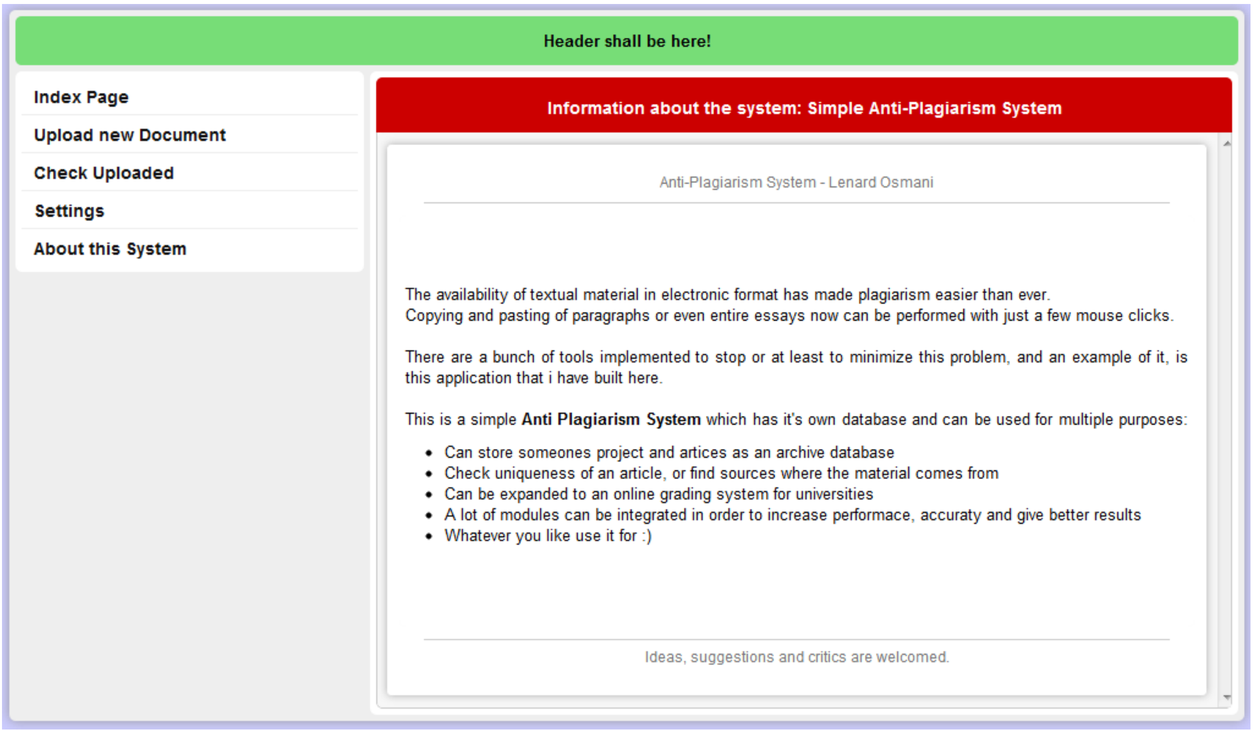
Pic.4: Example of the currently uploaded documents in the system



List of uploaded Documents

ID	Author	Title
7	Bad Student	Combination of some of the other algorithms (copied)
6	Wikipedia	Genetic Algorithm
5	Wikipedia	Tabu Search
4	Wikipedia	Particle Swarm Optimisation
3	Some Author	Some Meaningless Content
2	Some Author	Presidenti, dështim i politikës
1	Some Author	Leka Zogu dhe Sabri Godo, të ndryshëm dhe njësoj!

Pic.5: The “About this project” page.



Information about the system: Simple Anti-Plagiarism System

Anti-Plagiarism System - Lenard Osmani

The availability of textual material in electronic format has made plagiarism easier than ever.
Copying and pasting of paragraphs or even entire essays now can be performed with just a few mouse clicks.

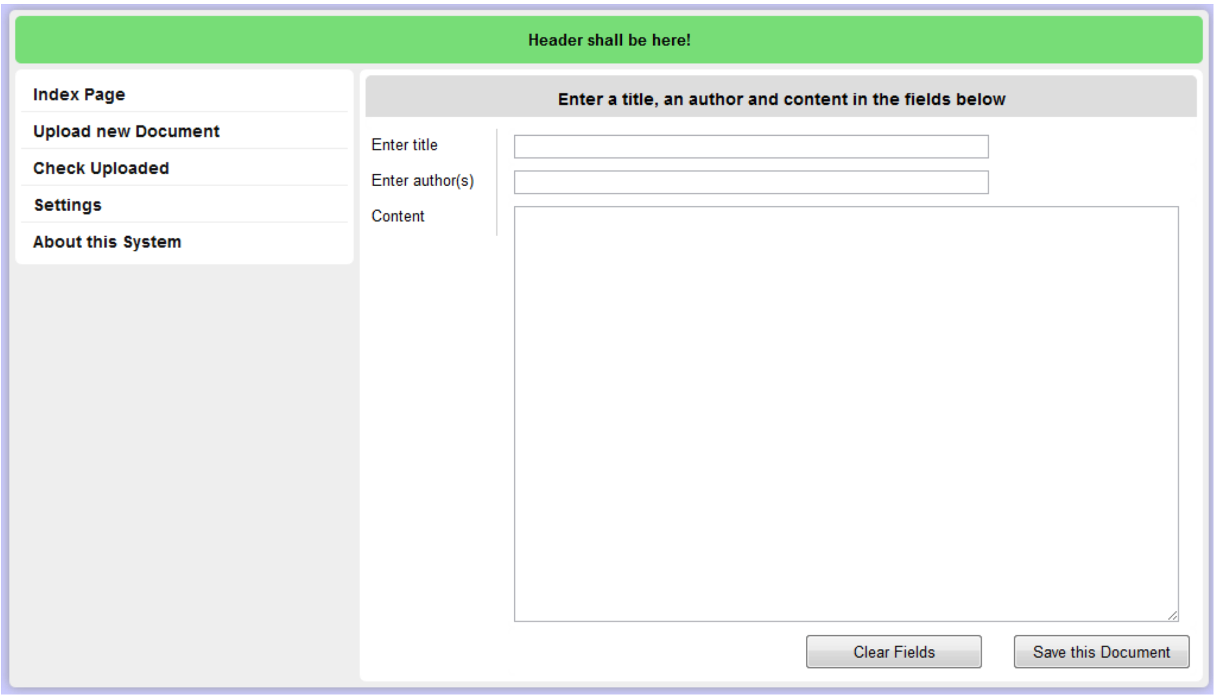
There are a bunch of tools implemented to stop or at least to minimize this problem, and an example of it, is this application that i have built here.

This is a simple **Anti Plagiarism System** which has it's own database and can be used for multiple purposes:

- Can store someones project and artices as an archive database
- Check uniqueness of an article, or find sources where the material comes from
- Can be expanded to an online grading system for universities
- A lot of modules can be integrated in order to increase performace, accuraty and give better results
- Whatever you like use it for :)

Ideas, suggestions and critics are welcomed.

Pic.6: The upload document page (limited functionality for now)



4.2 Algorithm and inner workings

Starting from the simple step of uploading a document, up to getting the final results for originality check is a long and complex process.

It goes through different steps which require specific algorithms and procedures to store, compare, update and print the results so the user can see them.

The list of steps is here:

1. **Upload document or enter content**
2. **Data Mining**

i. Interpret the content, divide it into small chunks of paragraph or sentences

ii. Analyze each chunk, and get a fingerprint of it

iii. Analyze structure of sentences, get weight and frequency words used (independent of language)

3. **Store in the Data-Base**

i. Store full document

ii. Store all chunks in chunks table

iii. Store sentence or paragraphs separately for display in case of detection

4. **Compare on the fly**

i. For every chunk or sentence, get the closest matching chunk from other documents

ii. Store in results table and link to current document through foreign keys

5. After finishing, get all results and print them highlighted aside the document viewer

4.3 Supported ways of detection

As we can see from the above section, we divide the document into different small pieces and for each of them we store different values.

First there is the complete sentence, then we have a fingerprint, a classification of language and words used and weight of each word, and finally there is the uniqueness index which can be the soundex of the words, or a custom hash function.

In my case, I use both soundex and a simple hash function.

All this different criteria are calculated and stored in order to offer us much more flexibility and more search criteria. Each of them can be used to get similar or copied results, but each of them has advantages and disadvantages.

For example, using the full sentence, gives us very exact and accurate results, but it will probably not detect copy if the user changed the text by using synonyms of the word, thus giving in smaller result set.

By using the finger print of the paragraph, the result might not be 100% accurate, (it takes similarity index which can be set as desired), but returns a much broader result set.

While using the uniqueness index, is a middle way, giving reasonable result set at a reasonable similarity index, but it might reveal not to be language independent.

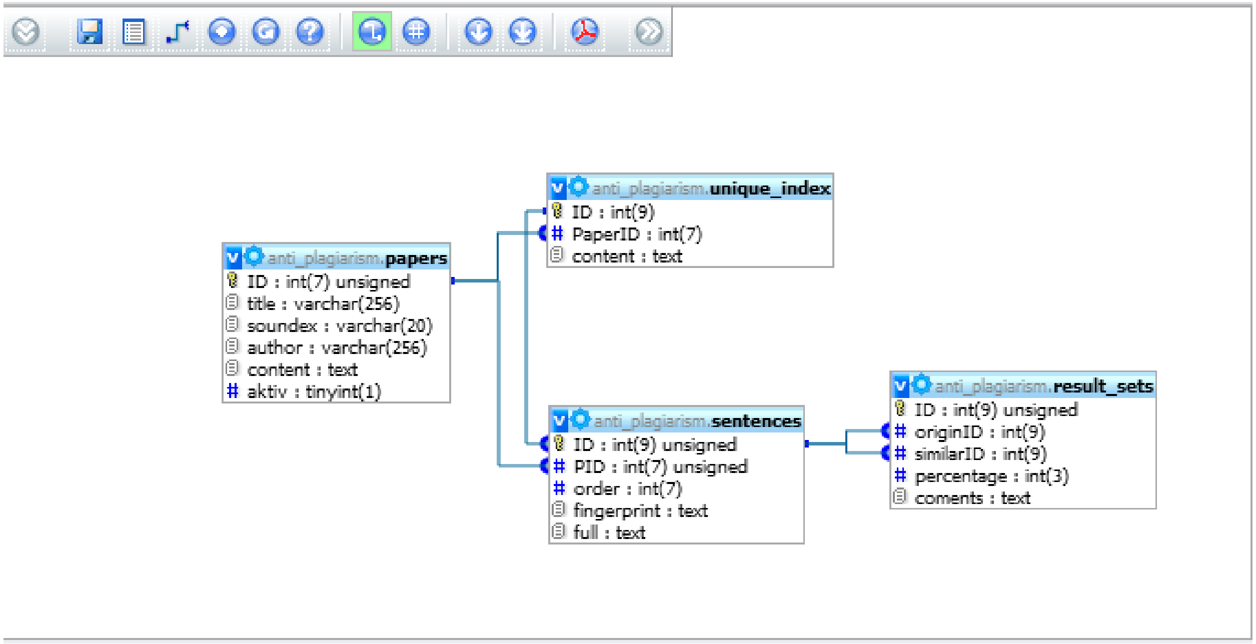
So, whichever way we use to detect plagiarism, has its advantages and disadvantages but are suited for different use cases.

4.4 Database

Up to this point, the database is pretty simple and has a small number of tables. It needs to store full documents, paragraphs, fingerprints and result so it is composed of 4 tables.

But the performance of the system is not limited by the tables, but by the number of documents uploaded or web articles crawled and stored. The more the system is used, the better and more accurate it gets, and result sets are more wide and complete.

Pic. 7 Shows the structure of the tables in DB and their relation



anti_plagiarism.papers
ID : int(7) unsigned
title : varchar(256)
soundex : varchar(20)
author : varchar(256)
content : text
aktiv : tinyint(1)

anti_plagiarism.unique_index
ID : int(9)
PaperID : int(7)
content : text

anti_plagiarism.sentences
ID : int(9) unsigned
PID : int(7) unsigned
order : int(7)
fingerprint : text
full : text

anti_plagiarism.result_sets
ID : int(9) unsigned
originID : int(9)
similarID : int(9)
percentage : int(3)
coments : text

4.5 Using the web-crawler to extend database

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Other terms for Web crawlers are ants, automatic indexers, bots, Web spiders, Web robots, or community Web scutters. [Wikipedia.]

This process is called Web crawling or spidering. Many sites, in particular search engines, use spidering as a means of providing up-to-date data. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches. Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses (usually for sending spam).

A Web crawler is one type of bot, or software agent. In general, it starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies.

The large volume implies that the crawler can only download limited number of the Web pages within a given time, so it needs to prioritize its downloads. The high rate of change implies that the pages might have already been updated or even deleted.

The number of possible crawlable URLs being generated by server-side software has also made it difficult for web crawlers to avoid retrieving duplicate content. Endless combinations of HTTP GET (URL-based) parameters exist, of which only a small selection will actually return unique content. For example, a simple online photo gallery may offer three options to users, as specified through HTTP GET parameters in the URL. If there exist four ways to sort images, three choices of thumbnail size, two file formats, and an option to disable user-provided content, then the same set of content can be accessed with 48 different URLs, all of which may be linked on the site. This mathematical combination creates a problem for crawlers, as they must sort through endless combinations of relatively minor scripted changes in order to retrieve unique content.

To do this task, I use a open source web crawler, which is build by Troy Wolf and is available onlie at http://www.troywolf.com/.

It basically is a proxy, or a simple page scraper, but it can efficiently extract content from given links or websites. With some small modifications, it can be converted in a web crawler which follows links embedded in a website, and goes in deeper levels of our specification.

After extracting content, it is stored in the DB where the headings can be used as title for the content, os we can simply use the URL, which should be displayed in the result set.

4.6 Garbage collector

With the passing of time, and sage of the system, the DB will grow larger and there will be plenty of (maybe) unused rows of data.

To lighten our DB without a penalty in accuracy and performance, a **garbage collector** might be needed.

There can be different strategies to how it might work and this is a very broad discussion topic, but the probably best strategy is to prioritize content that will probably not be needed.

For example we can eliminate the very old content, as through advances in technology it will probably get outdated and people are less likely to copy from that source.

Duplicate content (from different sources) should probably be discarded, because we already have the earliest origin of that information.

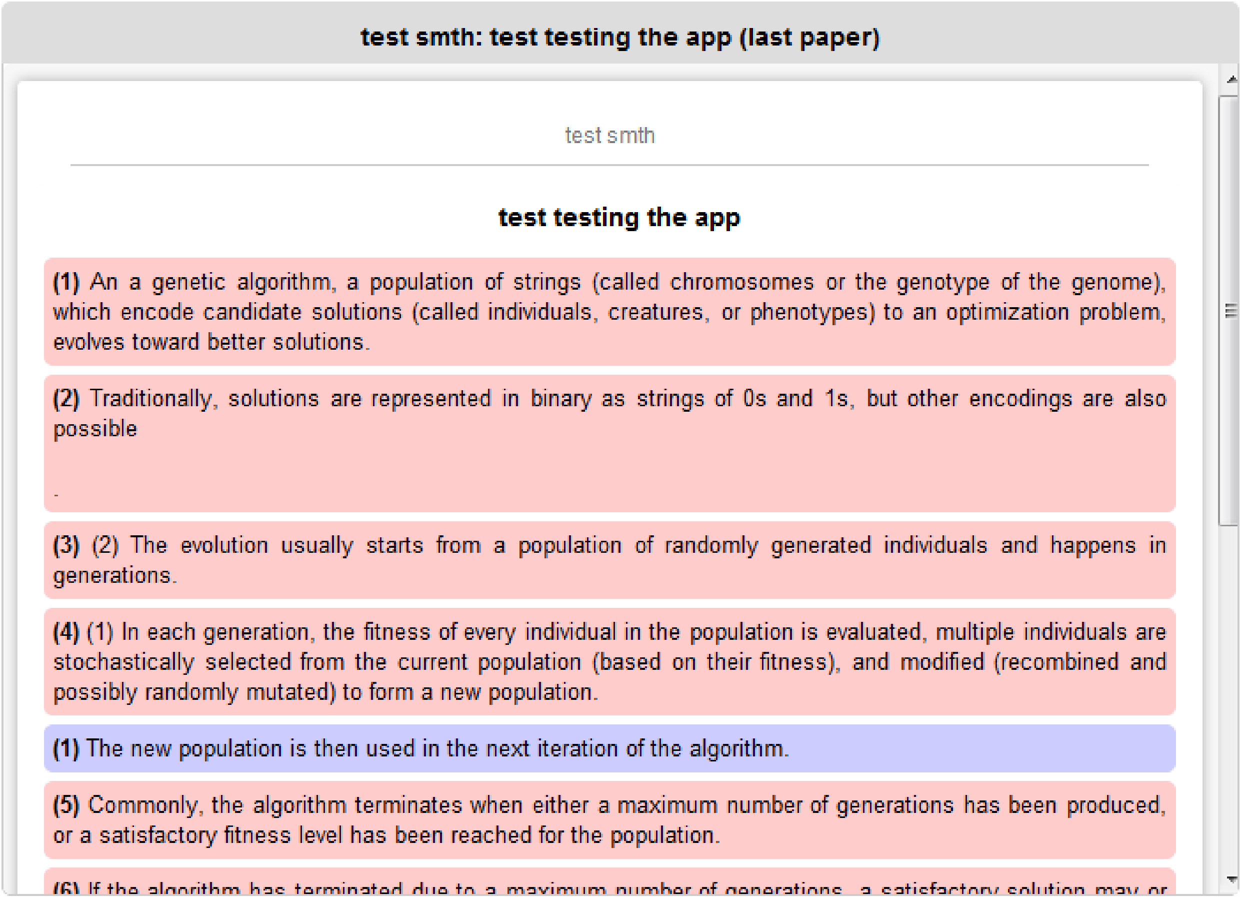
Or in case of websites, there can be priorities based on the topics or types of content that they publish. Popularity of a website, makes it a likely target for plagiarism, so unpopular or unupdated sites can be viewed as low risk for plagiarism.

4.7 Document viewer

This system stores the original documents or creates new ones on the fly in case that the content is entered through a simple form with some fields, and everything can be exported in different formats, but it also has a simple on-site document viewer for ease of use.

It has a simple and clear design and also highlights copied or similar content.

Pic.8. Shows the on-site document viewer.



test smth: test testing the app (last paper)

test smth

test testing the app

(1) An a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions.

(2) Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible

.

(3) (2) The evolution usually starts from a population of randomly generated individuals and happens in generations.

(4) (1) In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population.

(1) The new population is then used in the next iteration of the algorithm.

(5) Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

(6) If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or

4.8 Final tweaks

There is a setting page for this application, where the user can select some features of how the system should behave.

The user can choose:

1. **On which criteria is comparison based (full paragraph, unique index, fingerprint)**
2. **How many results to show for every chunk of data or whole document**
3. **Similarity percentage (how many % is a content considered as copied and how many % it is considered as similar, to detect synonyms or similar words)**
4. **The colors of highlighted fields**
5. **Allow export and download of papers and similarity result sets etc.**

CHAPTER 5

CONCLUSIONS & FUTURE WORK

5.1 – Conclusions: Even though the students are getting smarter in finding new complicated ways of plagiarism, and lack education not to do it, we still can do plenty of things to prevent them from doing so and also to detect when copyrights are being infringed.

The existing resources can give us a good place to start in order to educate ourselves and others about how to reduce this process, and give us pretty powerful tools how to detect them.

One such a tool was built to conclude this study, and prove that we have the means and technology to create constructive software.

5.2 – Future Work: Focus on expanding functionality to support more modules and file types (information sources) and building a larger dataset in order to get more complete and accurate results.

An important further development could be adding support for plagiarism detection not only on academic research, but on source code for software or specific projects (and image recognition).

REFERENCES

[1] Baker B.S. On finding duplication and near-duplication in large software systems // The Second Working Conference on Reverse Engineering, Toronto, Canada, 14-16 July, 1995. – Washington: IEEE Computer Society, 1995. – P. 86.

[2] Barnhart, R.K. (Ed.) Chambers Dictionary of Etymology – Edinburgh: Chambers, 1988. – 1284 p.

[3] Brown, L. (Ed.) the New Shorter Oxford Dictionary on Historical Principles – Oxford: Clarendon Press, 1993. – 3801 p.

[4] Ceska Z. The Future of Copy Detection Techniques // The 1st Young Researchers Conference on Applied Sciences, Pilsen, Czech Republic, 13 November, 2007. – Pilsen: University of West Bohemi, 2007. – P. 5-10.

[5] Clough P. Plagiarism in natural and programming languages: an overview of current tools and technologies // The 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, 13-17 March, 2005. – New York: ACM, 2005. – P. 776-781.

[6] EVE: Plagiarism Detection System. USA, 2000. [Online].Available: http://www.canexus.com/eve/index.shtml.

[7] Hanks, P. (Ed.) Collins Dictionary of the English Language. – London: Collins, 1979. – 1690 p.

[8] Harvey J., Robson S. The Accidental Plagiarist: An institutional approach to distinguishing between a deliberate attempt to deceive and poor academic practice // 2nd International Plagiarism Conference, Gateshead, UK, 19-21 June, 2006. – Newcastle: Northumbria University Press, 2006. – P. 16.

[9] Jadalla A., Elnagar A. PDE4Java: Plagiarism Detection Engine for Java source code: a clustering approach // International Journal of Business Intelligence and Data Mining – Vol. 3, No. 2 (2008), P. 121-135.

[10] Karp R.M., Rabin M.O. Efficient randomized pattern matching algorithms // IBM Journal of Research and Development – Vol. 31, No. 2 (1987), P. 249-260.

[11] Park C. In Other (People's) Words: plagiarism by university students - literature and lessons // Assessment and Evaluation in Higher Education – Vol. 28, No. 5, October 2003, P. 471-488.

[12] Plagiarism detection free detectors at wordchecksystems.com. [Online]. Available: http://www.wordchecksystems.com.

[13] Prechelt L., Malpohl G., Phlippsen M. Finding Plagiarisms among a Set of Programs with JPlag // Journal of Universal Computer Science – Vol. 8, No. 11 (2002), P. 1016-1038.

[14] Rehurek R. Semantic-based plagiarism detection: PhD Thesis Proposal – 2007, P. 6-13.

[15] Sorokina D., Gehrke J., Simeon W., Ginsparg P. Plagiarism Detection in arXiv // The Sixth International Conference on Data Mining, Hong Kong, Japan, 18-22 December, 2006. – Washington: IEEE Computer Society, 2006. – P. 1070-1075.

[16] The plagiarism resource site. Charlottesville: Lou Bloomfield, 1997. [Online]. Available: http://plagiarism.phys.virginia.edu/Wsoftware.html.

[17] Turnitin: Plagiarism Checker to Ensure Academic Integrity. San Francisco: iParadigsm, 1998. [Online]. Available: http://www.turnitin.com/static/index.html.

[18] TurnitinUK. New Castle: iParadigsm, 2010. [Online]. Available: http://www.submit.ac.uk/static_jisc/ac_uk_index.html.

[19] Wise M.J. String similarity via greedy string tiling and running Karp-Rabin matching: Technical report No. 463 – The University of Sydney, March, 1993. – P. 3-8.

[20] [22] Advego Plagiatus - Russia, Advego, 2008. [Online]. Available: http://advego.ru/plagiatus/. [Accessed: June 1, 2010].

[21] Scientific Journal of Riga, Technical University Computer Science. Information Technology and Management Science: Page 141 – 144, Volume 44, 2010.

[23] Wikipedia, The free encyclopedia, Available: http://www.wikipedia.com