Plagiarism Detection in Obfuscated Documents Using an N-gram Technique

Tomáš Kučečka*
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
kucecka.t@gmail.com

Abstract

Plagiarism is considered as a major problem these days especially in case of academic institutions. Very often students present someone else's work as their own and they are given credit for it. Therefore, we have to integrate plagiarism checking into the submission process which includes usage of plagiarism detection software. In this paper we focus on vulnerabilities of this software from the view of different obfuscations. We briefly introduce our statistical approach that should quite well detect obfuscated documents in case larger portion of the document is obfuscated. We show limits of this method and test effectiveness of n-gram technique used for detecting obfuscated documents. By playing with various parameters when dividing text into n-grams, we test success of n-gram method on similarity detection. We try to find the best parameters for the n-gram technique in order to achieve best detection results. We present the carried experiments and derive the conclusions.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous; I.2.7 [**Pattern Recognition**]: Natural Language Processing—*text analyses*

Keywords

plagiarism detection, obfuscating plagiarism detection, n-gram comparison

*Doctoral degree study programme in field Artificial Intelligence. Supervisor: Dr. Daniela Chudá, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, STU in Bratislava. Work described in this paper was presented at the 7th Student Research Conference in Informatics and Information Technologies IIT.SRC 2011.

Kučečka, T. Plagiarism Detection in Obfuscated Documents Using an N-gram Technique. Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies, Vol. 3, No. 2 (2011) 67-71

1. Introduction

The term plagiarism has numerous definitions, which basically means that two different people can mean something else by referring to this term. For instance, someone considers as plagiarism only a physical document that is a copy of someone else's work, while another can think of it as stealing ideas and thoughts from somebody. For this reason, various definitions of term plagiarism exist, for instance *"theft of intellectual property"* [8] or *"work of someone else is reproduced without acknowledging the source"* [5]. As we focus only on text documents in this article, under the term plagiarism we mean creating plagiary of author craft where by author craft we are primary interested in textual documents written in natural language.

Plagiarism detection plays an important role, especially in case of academic institutions. Using plagiarism detection systems in these institutions is crucial in order to keep some standards. This is not an easy task due to the amount of electronic data available on the Internet and easy access to them. Electronic documents can be easily copied or modified. There even exist special web sites called paper mills from which students can obtain pre-written papers. Therefore, plagiarism identification is a challenging task and cannot be addressed easily. Although various plagiarism detection systems are available, we definitely cannot say that these systems are able to detect all the plagiarism in a given corpus of documents. Actually, we are not even able to say how much of the plagiarised work was detected by the used plagiarism detection software.

The questions are: *Can documents that passed the checking be really considered as an original or not? How to successfully detect plagiarism and how can be this detection attacked?* In this paper we refer to our previous work in which we proposed a statistical method for detection of obfuscated documents. Although this method is very easy to implement and fast to run, its ability to detect obfuscation is vulnerable to portion of obfuscated text in a document. Therefore, in this paper we focuson effectiveness of n-gram technique used for checking documents on plagiarism. This technique is used by many of the available plagiarism detection systems which is the reason for its selection. From the view of our experiments, we are only interested in natural language documents written in Slovak language.

This paper is organized as follows. Section 2 describes related work in the field of plagiarism detection using the n-gram technique and document obfuscation. In sections 3 we introduce four obfuscation categories and propose basic methods that can successfully handle them. Section 4 shortly introduces out statistical approach to document obfuscation recognition and describes n-gram technique. Carried experiments with their results are summarized in section 5. Section 6 concludes this paper and proposes future work.

2. Related work

Articles [5, 2] give overview of an existing plagiarism detection tools, covering both source code and natural language plagiarism. These articles describe individual systems and approaches to plagiarism detection that are commonly used for detecting plagiarism and we recommend them as a good start for introduction in this field. Article [4] provides both the student's and the teacher's view on plagiarism detection.

N-gram approach to plagiarism detection is commonly used by many of the existing plagiarism detection tools. In [6, 7] an ENCOPLOT system is described which uses 16-gram and encoplot diagram to identify plagiarised documents.Different approach is used in [1] where suspicios document is split into sentences and this sentences are split into n-grams.

Plagiarism obfuscation means to deliberately perform such activity that would prevent plagiarism detection system to identify plagiarised text. In [10] a system AntiPlagiatKiller is introduced that performs different obfuscations. It uses two methods: white link-character insertion and letter substitution. Article [9] focuses on plagiarism detection from the view of using fingerprinting method. It proposes a new obfuscation-resistant approach called text sifting. It describes resistance of such method to cosmetic, scrambling and large-scale obfuscation.

3. Obfuscating plagiarism detection

Plagiarism check is usually performed on a large volume of data. For this, various detection systems have to be used as humans are not able to manually check every one submitted work. Plagiarised document can be hard to detect even if significant part of the document has been copied and consequently modified to hide this crime. Therefore, an important part of every plagiarism detection system is the built-in method that is used to find similarities between documents. If attacker knows details about this method, it makes the whole system vulnerable.

Always, the intent of an attacker is to modify the document such way that it is not identified as plagiary when compared to the original version (its source). To achieve this, attacker has several choices of modifying the document, for instance rewriting it in his own words. For some of the obfuscations there exists only little or even no defence, for instance paraphrasing detection is considered as a very challenging task. Here, we mention the most common methods that are used by attackers and discuss them.

At this point we suppose that the persons submitting their work are aware of it being checked by a plagiarism detection system. Therefore, we can expect some of them to deliberately modify it in order to pass this checking. We do not assume obfuscations that will make the document physically unreadable for the detection system. For instance encrypting the document or converting its text into an image. Also we suppose that the attacker has no access to the detection system, nor to the files created by this system after submitting the document on checking (for instance access to the hashed values when the detection system uses fingerprinting analyses technique).

We divided document obfuscation into the following four categories

- Cosmetic obfuscation
- Paraphrasing
- Scrambling obfuscation
- Shuffling obfuscation

We now describe individual categories in more detail and introduce possible solutions that should help to identify these obfuscated documents when submitted to the system.

Cosmetic obfuscation includes letter capitalization, changing formatting or adding punctuation to a plagiarised text. This category also covers modifications like replacing digit 2 for its Roman equivalent II. These obfuscations have effect when document is checked optically by a human eye but mostly none or only little effect when performing check via plagiarism detection system. That is due to preprocessing phrase in which generally the text is transformed to its plain format and optionally diacritic, capitalization, etc. is removed. Therefore, cosmetic obfuscation has almost none effect on comparison results.

Paraphrasing is the most common and widely used method that is very difficult if not even impossible to identify. More about paraphrasing strategies can be found in [1]. Paraphrasing identification is extremely difficult. In the preprocessing phrase it should involve usage of following strategies

- synonyms replacement - replacing syntactically different words with the same meaning with their one equal form.
- lemmatisation - process of determining word lemma where lemma is the canonical form of a word. Lemma can be obtained only from dictionary and depends also on the context of the text. For instance, words go, gone and went have the same lemma go.
- stop words removal - stop words are commonly used words in natural language text. For instance, in Slovak language these words are "a", "alebo", "na". These words are usually filtered out from the text as they are considered useless.

Scrambling obfuscation includes changing word in a document by adding, deleting ore replacing letters in it. Such modification causes match fail when comparing hashes of originally two same strings one of which was modified by this obfuscation. For instance, comparing words computer with computor or comuter. In the first case we

changed the letter e to o, in the second case letter p was deleted. Exact matching will therefore fail. Levenshtein distance [9] is much better as it returns degree of similarity between two words - it measures how much work has to be done to transform one word to another. Another alternative is LCS [14] which means finding the longest common subsequence of two strings.

Shuffling obfuscation includes white character insertion or letter replacing with their visually equivalent form. White characters are used instead of spaces, in which case they can join couple of words. For instance

"Niektorí študenti podvádzajú pri písaní svojich prác." [In English: "Some students cheat when writing their thesis."] "NiektoríXštudentiXpodvádzajúXpriXpísaníXsvojichXprác." [In English: "SomeXstudentsXcheatXwhenXwritingXtheirXthesis."]

In this example we inserted letter X in the plagiarised sentence to deceive plagiarism detection system when comparing these two sentences. Another common obfuscation method belonging to shuffling methods is replacing letters with their visually equal form. This usually includes finding character in other languages (Latin, Russian, etc.) that has equal shape to the replaced one. For example digit 1 (one) looks almost exactly the same as letter l when Times New Roman italic font is used. For instance

"Mám rád čokoládu a mlieko." [In English: "I like milk and chocolate."] "Mám rád čoko1ádu a m1ieko." [In English: "I 1ike mi1k and choco1ate."]

In the second sentence we used letter 1 (digit) instead of an alphabet letter l. This obfuscation forces plagiarism detection system to generate different hash code for the two sentences. Shuffling works only with special software, for instance Microsoft Word or Open Office where the attacker has possibility to use various fonts and symbols and he can format them appropriately.

4. Obfuscation detection

Obfuscated documents can be detected using statistical approaches. In our previous research we proposed a statistical method for detecting the shuffling obfuscation. We showed the simplicity of this method as it was based on word length watching and letter occurrence counting. If the returned values highly differed from the statistics we reported the document to the user as suspicious. The achieved results were quite satisfactory for us as our method achieved precision 87.78% with recall 98.75% in case when about 60% of the document text was obfuscated. Nevertheless, the recall rapidly drops with decreasing portion of plagiarised text. Although the success of detection depends also on the type of obfuscation (for instance if the attacker uses letter 'a' or letter 'x' as a white-link character), we cannot assume that the attacker would use the easiest way for us. Therefore, in this article we focus on performance of n-gram technique when shuffling obfuscation is done on a plagiarised text of a document. Our choice was based on the fact, that most of the existing detection system's use this technique as their comparison method.

N-gram comparison technique is based on splitting the text into strings of length n starting at a given position in the text. The position of the next n-gram is usually calculated from the position of an actual n-gram shifted by a given offset. The parameter n is given on the input by the user. The offset value depends on the used n-gram division. For instance, if n-grams are created by concatenating words, the offset is the number of words skipped when building the next n-gram. If the n-gram is created by joining letters not considering the end of the word, the offset represents number of skipped letters. N-gram division can quite vary among existing detection tools as there exist various approaches of splitting the text into n-grams, which basically include:

- overlapping n-grams - every n-gram starts on such position, that it has some common substring with the previous n-gram(s) also considering the position of that substring. For instance, splitting the string ABCDEBHAAC into n-grams with parameter n=3 and o=1 (meaning number of skipped letters) creates a set of following n-grams: ABC, BCD, CDE, DEB, HAA, AAC
- non-overlapping n-grams - none of the n-grams created from the text share the exactly same substring on the same position.

We implemented n-gram comparison technique into an existing plagiarism detection system called PlaDeS [3]. We used equation (1) to measure similarity between documents A and B.

$$sim(A,B) = \frac{Gram(A) \bigcap Gram(B)}{Gram(B)} * 100, \qquad (1)$$

where function Gram(A) returns set of n-grams in document A (analogically for B). Document B represents in our case the plagiarised and obfuscated document.

5. Experiments

In this section we report on our results achieved in comparing obfuscated documents using the n-gram technique. We tested robustness of this technique against the three obfuscations. These obfuscations are similar to those used to evaluate our statistical approach. An Example of these obfuscations will be shown on the following sentence: "Snahou je dosiahnut' úplné a jasne definované požiadavky, pričom tento proces je časovo náročný." [In English: "Aim is to achieve complete and defined requirements but this process is time consuming."].

- Obfuscation 1: we replaced every letter l for number 1 in every word of a document. We did not replace the letter 'L' or 'ľ' for the letter 1 as they are not visually similar. The obfuscated sentence looks: "Snahou je dosiahnut' úp1né a jasne definované požiadavky, pričom tento proces je časovo náročný." [In English: "Aim is to achieve comp1ete and defined requirements but this process is time consuming."].
- Obfuscation 2: We replaced every fourth space separating two words with character 'X'. We gave constraint that one of the joined words is at least of length two. If the words were separated by a number of spaces, we replaced all of them. The obfuscated sentence looks: "Snahou je dosiahnut' úplnéXa

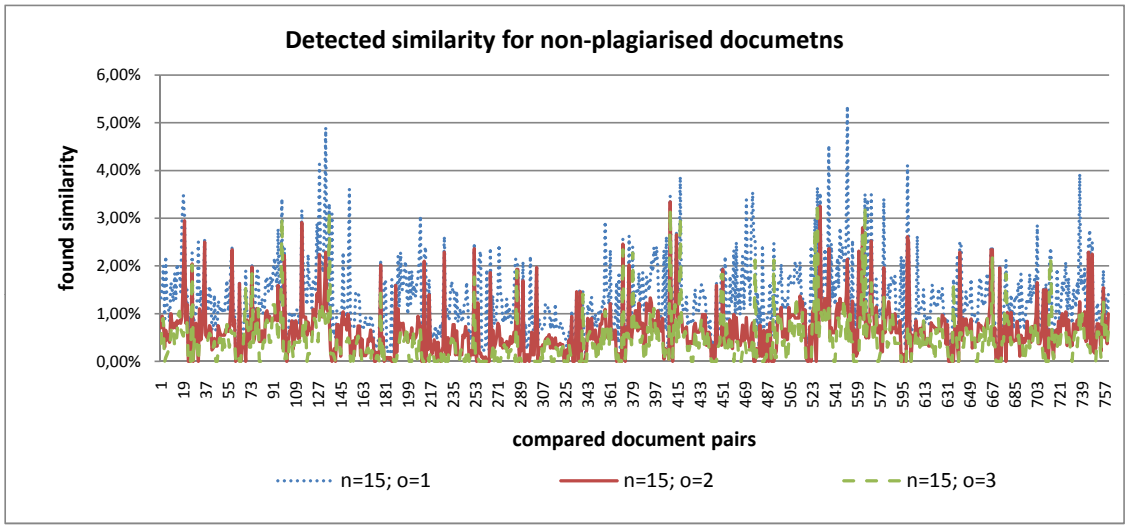

Figure 1: Detected similarity for the non-plagiarised pairs of documents. Parameter n is the n-gram length and the o parameter is the number of letters used as offset.

jasne definované požiadavkyXXpričom tento proces je časovoXnáročný." [In English: "Aim is to achieveXcomplete and defined requirementsXbut this process is time consuming."].

- Obfuscation 3: We randomly added or deleted letter from a middle of a word. We made sure that the word modified by this obfuscation is at least 5 characters long. The added letter was randomly generated from the set 'a-z'. The obfuscated sentence looks: "Snahou je dosiahnut' úplné a jasne definoevané požiadavky, pričom tento prces je časvo náročný." [In English: "Aim is to achieveacomplete and defined requirementsebut this process is timelconsuming."].

We obfuscated 20 documents with each of the three obfuscations (the total number of obfuscated documents was 60). The documents were selected randomly from the corpus of 283 student term papers. All papers were written in Slovak language. The average number of words on one document in the corpus was 2290. The portion of the obfuscated text in the document was 30% as we consider it as an interesting balance between detection problem and student's portion of plagiarised text. Input to the comparison process were the 20 obfuscated documents together with their original versions. Only the obfuscated text was included in the plagiarised document. This means that in case 30% of the document text was plagiarised (copied), the whole content of the plagiarised document sent to the detection system was the obfuscated text.

We focus on the following two parameters regarding n-gram technique

- n-gram length - in case of word n-grams it is the number of words forming the n-gram. In case of letter n-gram it is the number of letters.
- n-gram overlapping factor - created n-grams can overlap which means that they share a common substring

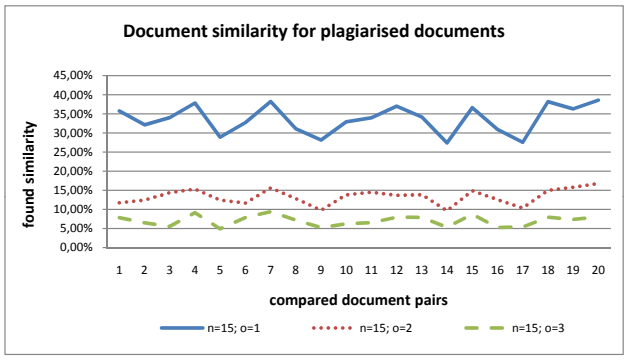

Figure 2: Detected similarity for the plagiarised pairs of documents on which second obfuscation was performed.

The comparison was evaluated one way - we watched what portion of obfuscated text is found in every source document. We experimented with the value of parameters n (n-gram length) and o (offset). We decided to set the length of an n-gram to the value 15 as average word length in a document written in Slovak language is 5.8 based on our previous experiments.

The detection of obfuscation 1 depends on the number of 'l' characters in the document. This number was quite small therefore the detection was quite successful with the similarity identification of about 90%. The obfuscation 2 showed to be much more dangerous as the original detection method based on word 3-grams returned 0% for the plagiarised couples. The offset in this case was one word. Therefore, we tried to evaluate the similarity with letter 15-grams using the various offset. Figure 1 shows the detected similarity for the plagiarised document couples and in figure 2 we give detected similarities for the document couples that are not similar.

The results show, that with the growing value of n-gram offset the found similarity drops together with the false identification of non-plagiarised documents. This was expected as with the smaller offset we are able to detect

more similar n-grams between the words concataneted with random letters. In our case these were the letters 'X'. The fact that we detect more similar n-grams that do not reflect document similarity is due to larger number of shared n-grams for parts of text that are between two documents common. This is the reason for growing inaccuracy.

The third obfuscation is very similar to the second one and does not mean any additional type of problem for the n-gram copamrison. The only difference is in position where the letters are replaced, added or deleted. Note that the presented n-gram technique for document comparison is able to detect similarity between obfuscated documents only if n letters plus the given offset were unchanged between two obfuscations.

6. Conclusion

In this paper we carried out experiments on plagiarism detection in obfuscated documents. The used comparison method was the n-gram technique. We did not focus on identification of the obfuscated documents but on determining similarity between the plagiarised parts of the document and the obfuscated ones. We created 20 plagiarised pairs for each obfuscation where the plagiary had 30% of the text from its original source. We obfuscated this text using the shuffling obfuscation. We experimented with various parameters of n-gram technique and showed its performance from the view of similarity detection. Overall, the detection based on letter division showed to be much more effective on discovering the similarity than in the case of using word division. While in the second case the detected similarity was about 0% when 3-grams were used, in the first case we managed to detect 33% similarity on average using the 15-grams divided by letters with the offset value equal 1.

Based on the experiments results we propose to change the similarity measure. In this article we computed the similarity as fraction of shared n-grams between two documents to the all extracted n-grams. The new similarity should be based on the fraction of shared letters to the all letters between two documents. We believe that this will enable us to increase the similarity percentage between the two documents, one of which is obfuscated. For instance, in case of inserting random letters in the middle of words from a document in order to obfuscate it, the only difference in the proposed similarity measure would be the count of additional letters in the denominator. Of course, we assume that the inserted letters are distant at least the length of an n-gram plus the offset from each other.

Acknowledgements. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0971/11 and by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 345-032STU-4/2010, and it is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] A. Barrón-Cedeño and P. Rosso. On automatic plagiarism detection based on n-grams comparison. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 696–700, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] S. Burrows, S. M. M. Tahaghoghi, and J. Zobel. Efficient plagiarism detection for large code repositories. *Softw. Pract. Exper.*, 37:151–175, February 2007.

[3] D. Chuda and P. Navrat. Support for checking plagiarism in e-learning. *Procedia - Social and Behavioral Sciences*, 2(2):3140 – 3144, 2010. Innovation and Creativity in Education.

[4] D. Chuda, P. Navrat, B. Kovacova, and P. Humay. The issue of (software) plagiarism: A student view. *Education, IEEE Transactions on*, PP(99):1, 2011.

[5] P. Clough. Plagiarism in natural and programming languages: an overview of current tools and technologies. *Finance*, (July):1–31, 2000.

[6] C. Grozea, C. Gehl, and M. Popescu. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship, and Social Software Misuse.*, page 10, 2009.

[7] C. Grozea and M. Popescu. Who's the thief? automatic detection of the direction of plagiarism. In *CICLing*, pages 700–710, 2010.

[8] R. Lukashenko, V. Graudina, and J. Grundspenkis. Computer-based plagiarism detection methods and tools: an overview. In *Proceedings of the 2007 international conference on Computer systems and technologies*, CompSysTech '07, pages 40:1–40:6, New York, NY, USA, 2007. ACM.

[9] M. Malkin and R. Venkatesan. Comparison of texts streams in the presence of mild adversaries. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research - Volume 44*, ACSW Frontiers '05, pages 179–186, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[10] P. Yurii. "counter plagiarism detection software" and "counter counter plagiarism detection". In *Proceedings of the 25th annual conference of the Spanish society for Natural Language processing*, pages 179–186, 2009.