intellectual system, searching for plagiarism, signatures, data space

Natalya SHAKHOVSKA[*]*, Iryna SHVOROB*[**]

THE METHOD FOR DETECTING PLAGIARISM IN A COLLECTION OF DOCUMENTS

Abstract

The development of the intelligent system for searching for plagiarism by combining two algorithms of searching fuzzy duplicate is considered in this article. This combining contributed to the high computational efficiency. Another advantage of the algorithm is its high efficiency when small-sized documents are compared. The practical use of the algorithm makes it possible to improve the quality of the detection of plagiarism. Also, this algorithm can be used in different systems text search.

1. INTRODUCTION

Nowadays, the Internet is the biggest source of information. Now, people can easily search, get access and browse the web to get the information they need. Just imagine how difficult it would be to do scientific research without the Internet and web space. Furthermore, due to the size and digital structure of the internet, it is easy to illegally use someone else's work now.

The problem of plagiarism has a direct relationship with the scientific community. The most common plagiarism is written text document which is formed by copying some or all parts of the original document, sometimes with some modifications. Identification of documents which were copied is stressful and time-consuming process to humans due to the large number of documents which have to be analyzed. The documents in digital format make the process of plagiarism quite simple, it means that such cases of plagiarism can be traced automatically.

[*] Lviv Polytechnic National University, Lviv, Ukraine, S. Bandery Str., 12, Lviv, 79013, natalya233@gmail.com

[**] Lviv Polytechnic National University, Lviv, Ukraine, S. Bandery Str., 12, Lviv, 79013, irka.shvorob@gmail.com

Plagiarism detection depends on many factors[5]. The first factor is the presentation of the document, which essentially covers the characteristics of the document as a preliminary step to compare [7, 8, 9]. These representations include model of identification tags, N-grams, probabilistic models, algorithms "scales" and others. Most of these representations work well in detecting verbatim plagiarism, but are vulnerable to identify complex patterns of plagiarism.

The second factor is a similarity and a measure of proximity, which is used to calculate the similarities or differences between sentences. Given the behavior of plagiarists, which usually includes insertion, deletion or substitution of words necessary to determine which activities are best for detection of plagiarism.

With the development of information systems the number of areas to identify plagiarism text only increased. This is the area of scientific papers, various publications in the field of journalism, fiction genres [13–15].

Currently, there are many methods and algorithms that can detect plagiarism of text objects. But over time, there are new challenges associated with the development of information systems. These tasks require more qualitative and more accurate detection of plagiarism in text.

Purpose of this paper is to improve the efficiency and quality of plagiarism detection in text objects by the use of the combined algorithm.

2. THE INTELLIGENT SYSTEM OF DETERMINE THE DEGREE OF RESEMBLANCE OF THE TEXTS

2.1. The algorithm development

Opt Freq algorithm implements the method of "optimal search frequency" and its used to search for similar documents in a wide range of applications, from web to clustering news. The gist of it is this. Instead of classical metrics TF*IDF a modified version of it is proposed. We introduce a heuristic concept of “optimal frequency” for the word “equal” $\ln\left(\frac{10}{1000000}\right) = 11.5$ which means “the optimal” entering of word in 10 documents from 1000000. If the real value of IDF is less than "optimal", then it slightly (by law parabola) rises to $IDF_{opt} = \sqrt{\frac{IDF}{11.5}}$, and if it is greater it significantly (as hyperbole) reduces to

$$IDF_{opt} = \sqrt{\frac{11.5}{IDF}} \quad (1)$$

For the collection the dictionary is created. This dictionary puts every word in accordance with the number of documents in which this word occurs at least once (df). Then the frequency dictionary for document is built and the "weight" wt of each word is calculated by the formula:

$$wt = TF * IDF_{opt}\ , \tag{2}$$

where

$$TF = 0.5 + 0.5 * \frac{tf}{tf_{max}}, \tag{3}$$

$$\text{IDF} == \log\left(\frac{\text{df}}{\text{N}}\right), \tag{4}$$

$$IDF_{opt} = \begin{cases} \sqrt{\frac{IDF}{11.5}}, IDF < 11.5 \\ \frac{11.5}{IDF}, IDF \ \geq 11.5 \end{cases} \tag{5}$$

tf (term frequency) is the ratio of occurrences of a word to the total number of words of the document. Thus, the estimated importance of words within a single document:

$$tf = \frac{n_i}{\sum_k n_k} \tag{6}$$

where n_i is the number using the word in a document, and the denominator – the total number of words in this document.

df (inverse document frequency) is inversion frequency with which a certain word is found in the documents collection. Consideration df reduces weight widely used words:

$$df = \log\frac{|T|}{|T_i \supset a_i|} \tag{7}$$

where $|T|$ is count of text documents in collection; $|T_i \supset a_i|$ is count of text documents, where word a_i occurs (where $n_i \neq 0$).

Then the 6 words with the largest values of wt are selected and concatenated in alphabetical order into the string. The check sum of the resulting line is calculated as the signature of document [3, 6].

Also, it is very important, where part of text is arisen [10, 11].

First of all, we introduce the concept of weight sentence.

$$Location = \frac{1}{n*m} \tag{8}$$

where $n = \overline{1..3}, m = \overline{1..3}$ – the place calls to the main part and paragraph respectively. Begin and end of text or paragraph estimated value of 1, the middle is as 3.Coefficient key phrase is determined by entering the sentence U of elements of a set of significant sentences from A membership function:

$$Cuephrase = \mu_A(U) \tag{9}$$

A={«Conclusion», «In the end», «By the way»…}.

Index of statistical significance is formed on the basis of visiting sentence key-words specified by the author of the article:

$$Statterm = \mu_K(U) \tag{10}$$

The value added is defined as the presence of terms related words sentences that appear in the article's headline to the total number of words in a sentence (words) except for words whose length is less than 3 characters:

$$Addterm = \frac{word}{words} \tag{11}$$

The weight of text block U is:

$$Weight(U) = Location(U) + Cuephrase(U) + \\ + Statterm(U) + Addterm(U) \tag{12}$$

So after being allowed to study all the documents necessary to accomplish the following: to exclude a statement that its content has hit the consolidated data repository and perform the final sorting sentences. For the task of bringing to the final ranking factor "information novelty" use the following method:

– Let we have two sets of sentences $B = \emptyset$ and $A = \{A_i | i = 1,2,\dots,N\}$, N is count of sentences in text. For every sentence A_i the usefulness $P(i)_i$ q_i:

$$P(i)_i = q_i,\ i = 1,2,\dots,N \tag{13}$$

– The sentences from set A sort Descending $P(i)_i$
– If A_i has the biggest $P(i)_i$, we take it in B. The usefulness for sentences in A set s

$$P(i) = \frac{P(i)}{kq_i} \tag{14}$$

where k > 0 – factor clipping similar sentences.

– Is A empty? If NOT, go to 1.

The next problem is information estimating from different sources [15–16]. For semi-structured data type text file with a known format – dictionary data types defined formatting released the text of the formatting, copying its contents:

$$object \rightarrow Find\left(\pi_{firmattype}\left(\sigma_{object}(Dic)\right)\right) \quad (15)$$

```
foreach object
Selection
. ParagraphFormat.Alignment = Left (1, formattype)
. Font.type = Mid (formattype, 3, 1)
. Font.Caps = Right (formattype, 1)
InStr (1,. Text, Right (formattype, 2);
Copy.Selection
```

2.2. The proposed algorithm

The proposed algorithm can be divided into the following stages.

1. The construction of the dictionary of words.

The vocabulary is created throughout the collection. Each word is associated with a number of documents that it occurs at least once (df) and the average length of the document is determined (dl_avg).

2. The construction of the frequency dictionary.

The frequency dictionary to document is constructed and to each word its "weight» wt on a formula Okapi BM25 with parameters k = 2 and b = 0.75 is calculated.

3. The construction of signatures.

11 I-Match signatures created for each document. The basic idea of this approach is to calculate the daktilohra I-Match for presentation of the documents content. For this purpose, initially for a collection of documents created dictionary L, which include words with average values IDF, because these words provide are usually more accurate results in identifying fuzzy duplicates. Words with large and small values IDF rejected.

Then, for each document set of different words U is formed belonging to it, and determined the intersection of U and L. If the size of the crossing of vocabulary over some minimum threshold (determined experimentally), the list of words that are included into intersection are ordered, and it is calculated I-Match signature (hash function SHA1). In addition to the main dictionary L created K various dictionaries L1-LK, obtained by accidental deletion from the initial dictionary some fixed small part of words p, part of about 30%–35% of the original amount L.

For each document, instead of one calculated (K +1) I-Match signature by the described above algorithm, the document that is represented as a vector of dimension (K+1) and two documents are considered duplicates if they match at least one of the coordinates.

If the document undergoes small changes (order n words), then chances are that there are at least one of K additional signature remains unchanged, is:

1-(1-pn)K (*)

Indeed, the likelihood that change will not affect any other dictionary is pn - the probability that all changes fall into a remote part of the original dictionary. Then (1-pn) - the probability that the signature will change, and (1-pn)K - the probability that all signatures will change (because additional dictionaries formed independently), so (*) - is the desired probability.

Recommended values of parameters that are well proved in practice p = 0.33 and K = 10.

4. The find of duplicate.

Documents which have at least one coincidence of signatures considered as duplicate (There is a conflict of hash codes).

3. THE SYSTEM ARCHITECTURE

To build an information system model is used CASE-tool AllFusion Erwin Data Modeler , which enables model based infological model of information system build its datalogical model and create a database in any database management system. The development of the summarization system provides in the notation IDEF1X.

During the implementation of systems analysis for this area following charts were developed[4]:

1. IDEF0-diagram for subtasks of the main business process (figure 1);
2. IDEF3-diagram for the block "Choosing the algorithm of working with words" (figure 2).

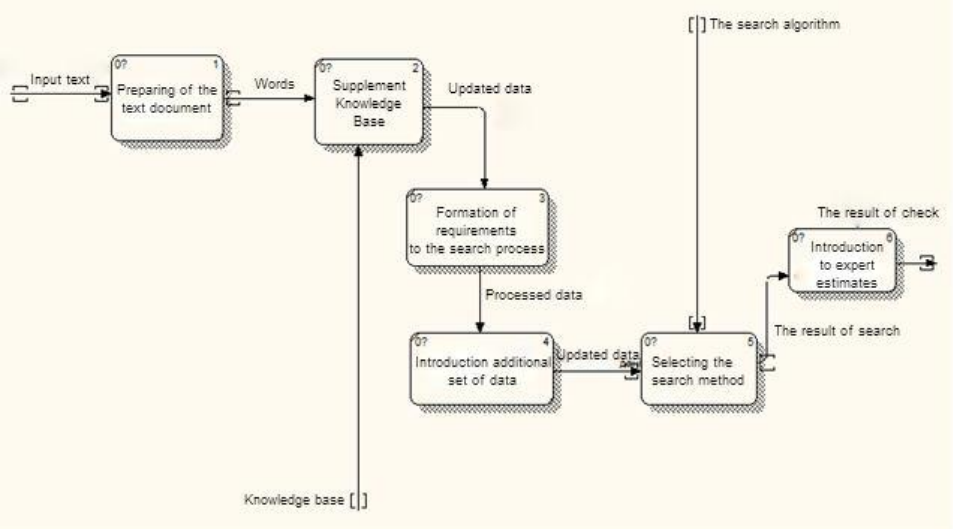

Fig. 1. IDEF0-diagram for subtasks of the main business process [own study]

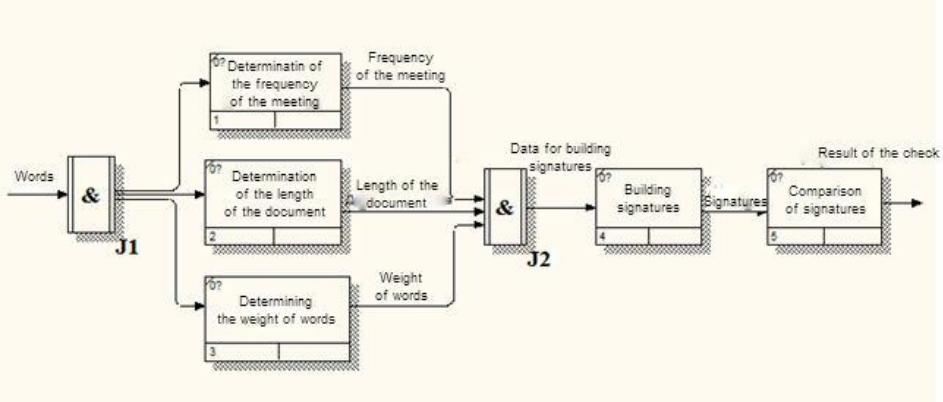

Fig. 2. IDEF3-diagram for the block "Choosing the algorithm of working with words"[own study]

The work "Checking for plagiarism in the text" is divided into 6 works: "Preparing of the text document", "Supplement Knowledge Base", "Formation of requirements to the search process", "Introduction additional set of data", "Selecting the search method", "Introduction to expert estimates". These works are carried out in the system sequentially, one after another. A text document which gets into the system due to user actions is applied to the input to the work "Preparation of the text document". Text information namely data entered in the system after this work is the result and therefore the input information for the work "Supplement knowledge base". This information is converted into data

format suitable for the system in which they are ready for further processing. Checking words of text is carried out as a result of the "Supplement Knowledge Base". The result of this work is a set of sentences which will be applied to the input of the" Formation of requirements to the search process" for further processing and the input of "Introduction additional set of data". These works are carried out the text processing. Words and formed signatures are the result of the work "The introduction additional set of data". They apply to the input of "Selecting the search method" and then searching for plagiarism is carried out. The next work "Introduction to expert estimates" provides the end result – a numeric value of the searching for plagiarism.

The IDEF3-diagram for the block "Choice of algorithm with the words" is consists of such units of work: "Determination of the frequency of the meeting" (determines the number of meeting of words in the text, returns the number of meeting of words in the document), "Determination of the length of the document" (determines the length of the document), "Determining the weight of words" (the data obtained in previous studies and knowledge base are used and keywords of the text are assigned of weight), "Building signatures" (connecting words into signatures), "Comparison of signatures" (checking of signatures, a collision of hash codes takes place).

The proposed system has a large number of works. These works are different, and usually independent. So, for greater flexibility they should be divided on the modules.

The system consists of the following modules:

- the database and knowledge;
- the subsystem of integration and information gathering;
- the subsystem of analysis.

Database and knowledge is designed to collect structured data and meta data about used text. It is a central part of the program because it is used by all the other parts

Whereas the system provides integration with other systems and sources of research documents, it must have functionality that would allow it to read the files and render its data to other systems. The subsystem of integration and data collection corresponds for this.

The analysis is carried out based on the downloaded document. The subsystem of analysis implements the proposed document analysis algorithm.

Input data are:

1) file with the text for which the test will be performed. File format .doc or .docx;
2) file with the text, which will be carried out the test input document. File format: .doc or .docx.

The result of the algorithm is a numerical value of the degree of the resemblance of the texts. This refers to a real number within a [0, 1]. If the result returns 0, it means that the texts are different, if 1 - text completely identical.

There also conclusion will be displayed: if the numerical value of the degree of resemblance of the texts of more than 0.3, the selected text - plagiarism, if less than 0.3, it is not plagiarism.

Figures 3a and 3b shows the software implementation of the developed system and the results of its implementation.

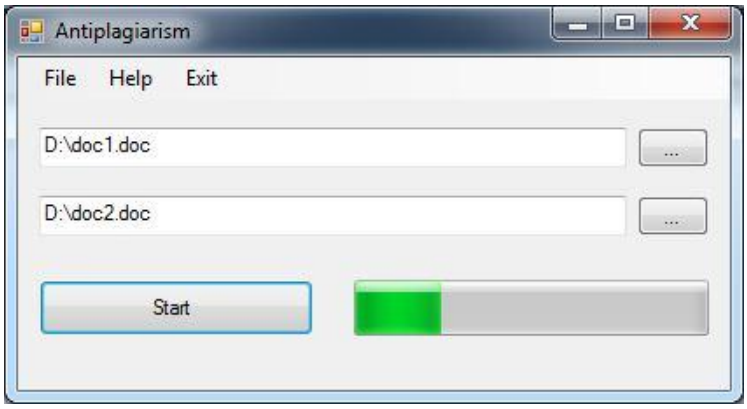

Fig. 3a. The test example of the program [own study]

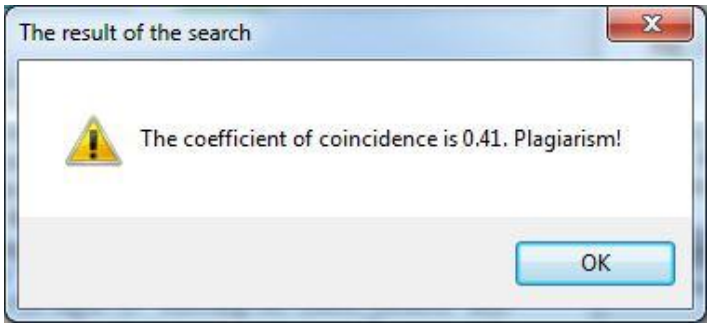

Fig. 3b. The results of the program [own study]

Before checking for plagiarism the system must makes its preliminary treatment.

This stage applies to all requested documents, as well as primary documents. There are four steps at this stage:

- remove stop words;
- fragmentation of the text;
- tokenization of the text;
- selection of roots of words.

4. CONCLUSION

The problem of plagiarism has been established and discussed. Two algorithms for finding fuzzy duplicates were considered and incorporated. The verification of the considered algorithms and combined algorithm was made. The received data are shown in Table 1. It is worth noting, as a result of combining it has improved its performance and result of searching of fuzzy duplicates.

System analysis for the intelligent system of determines the degree of resemblance of the texts was carried out and two charts were developed.

Basic steps for preprocessing the text were identified.

Tab. 1. The results of verification of algorithms [own study]

	Lex Rand Algorithm	Algorithm Opt Freq	Combined Algorithm
The accuracy of searching of plagiarism	39	59	61
A validation (sec.)	35	41	39

Obviously, the algorithm is not perfect in solving the problem of determining fuzzy duplicates. In order to improve options for combining multiple algorithms. For example, using the method of "descriptive words" can determine what class includes documents are scanned as each generated vector uniquely identifies this class. Then identify duplicates in a particular class of documents, signatures using methods based on the analysis of special characters. In this case, the possible increase effectiveness duplicate determination in a particular class of documents.

Duplication of texts in information flows is not always a negative phenomenon in terms of the user who uses the Internet for business purposes. An example of such an exception, for example, ranking brand when republication counts the number of press releases. Also you can use a number of overlapping signs "measure of importance" of a message and more.

REFERENCES

[1] PARK S.-T., Pennock D., LEE GILES C., KROVETZ R.: *Analysis of Lexical Signatures for Finding Lost or Related Documents*. Finland, 2002, p. 8.

[2] NIKOL'S'KIJ J., PASICHNIK V., SHHERBINA J.: *Sistemi shtuchnogo intelektu*. Vidavnictvo Magnolija-2006, L'viv 2010, p. 279.

[3] ZELENKOV J., SEGALOVICH I.: *Sravnitel'nyj analiz metodov opredelenija nechetkih dublikatov dlja Web-dokumentov*. Devjataja konferencija KSB, 2007.
[4] KATRENKO A.: *Systemnyj analiz: pidruchnyk z gryfom MON*. Magnolija-2006, L'viv 2009, p. 352.
[5] MAURER H., KAPPE F., ZAKA B.: *Plagiarism – A Survey*. Journal of Universal Computer Sciences, Vol. 12, No. 8, 2006, pp. 1050 – 1084.
[6] The Open Archives Initiative Protocol for Metadata Harvesting Protocol Version 2.0 of 2002-06-14. http://www.openarchives.org /OAI/2.0/openarchivesprotocol.htm
[7] HRYTSENKO V.: *Ynformatsyonnye tekhnolohy: tendentsyy, puty razvytyia*. Upravliaiushchye systemy u mashyny, No. 5, 2001, pp. 3-20.
[8] KRAIOVSKYI V.: *Osnovni pidkhody do rozroblennia prohramnoho kompleksu avtomatychnoho referuvannia tekstovykh dokumentiv* [in:] Kraiovskyi V. I., Lytvyn V. V., Shakhovska N. B., Zbirnyk naukovykh prats NAN Ukrainy, Instytut problem modeliuvannia v enerhetytsi, No. 51, Kyiv, 2009, pp. 178-186.
[9] PARK S.-T., PENNOCK D., LEE GILES C., KROVETZ R.: *Analysis of Lexical Signatures for Finding Lost or Related Documents*. Finland 2002, p. 8.
[10] KOŁCZ A., CHOWDHURY A., ALSPECTOR J.: *Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization*. KDD, 2004.
[11] BRODER A.: *Identifying and Filtering Near-Duplicate Documents*. COM'00, Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, 2000, pp. 1-10.
[12] BERSON T.: *Differential Cryptanalysis Mod 232 with Applications to MD5*. EUROCRYPT. http://dl.acm.org/citation.cfm?id=1754956.
[13] HAHN U., MANI I: *The Challenges of Automatic Summarization*. Computer, Vol.33, No. 11, 2000, pp. 29–36.
[14] Document Understanding Conferences (DUC): Web site, 2008. http://duc.nist.gov. 15.10.2011.
[15] ALYIGULIEV R.: *Avtomaticheskoe referirovanie dokumentov s izvlecheniem informativnyih predlozheniy*. Vyichislitelnyie tehnologii, Vol. 12, No. 5, 2007, pp. 5-15.
[16] YANG C., WANG F.: *Fractal Summarization for Mobile Devices to Access Large Documents on the Web*. Proc. of the WWW2003, May 20-24, 2003, Budapest, pp. 26-31.