Computational load reduction of fuzzy duplicate detection in large amounts of information

To cite this article: E Sharapova 2020 IOP Conf. Ser.: Mater. Sci. Eng. 734 012119

View the article online for updates and enhancements.
Computational load reduction of fuzzy duplicate detection in large amounts of information

E Sharapova
Vladimir State University, 23, Orlovskaya str., Murom, 602264, Russia

E-mail: mivlgu@mail.ru

Abstract. The paper deals with the detection of fuzzy duplicates of documents in large amounts of information with low computational costs. The existing methods give either low search completeness at low computational costs, or acceptable completeness at very large computational costs. It is proposed to use combined method of detecting fuzzy duplicates. At the beginning of the whole set of documents with the help of signatures similar texts are searched and then, using context analysis methods, a detailed comparison of the texts found in this way is carried out. The method first performs an approximate search for similar documents using description words signature with a small match threshold. A detailed search for matches in previously found documents is performed using the shingles method.

1. Introduction
Development of information technology and the Internet gave a wide range of users access to large volumes of information. There are large counts of online libraries with electronically literature. It became possible to read books, newspapers and news directly from the computer screen. In Internet there are many methodical literature, lectures, textbooks, etc. There are big collections of essays, laboratory work, course and diploma projects and even dissertations. The using of computer technology greatly facilitated the task of finding and copying of such information.

The technique of simple copying information from one or more sources with minimum editing text has name «Copy & Paste». This technique is greatly used for writing papers. Thus, there is the problem of detecting texts compiled from other documents.

One text document may be a part of another document; formed from several documents; may be modified part of another document (with a change of words to synonyms, endings, times, etc.).

Fuzzy duplicates are texts that have undergone some adjustments and modifications, but retain a semantic similarity with the originals [1].

There is a wide range of tasks that require the detection of texts with similar content. Systems of plagiarism search are widely used in publishing houses and educational institutions. Systems for checking the uniqueness of texts are used when ordering the services of copywriters. News aggregators identify similar news, group them into single chains and remove duplicates. Anti-spam systems track the occurrence of a large number of emails with similar content. For this reason, the task of detecting texts that are significantly similar to each other (fuzzy duplicates) is quite relevant.

Of interest is the use of approaches that allow searching for similar documents with a minimum number of comparison operations.
2. Methods of fuzzy duplicate detection

There are two large groups of the most popular methods for detecting similar texts (fuzzy duplicates): content analysis and signatures. The first group involves an analysis of the similarity of the content of the texts. Various methods can be used for these purposes. Often used such classic approach as boolean model, vector space model and bags of words [2, 3]. These approaches analyze the dictionaries intersection of checked texts without taking into account the order of words in sentences. On the one hand, this gives good performance, on the other, it does not allow to determine the percentage similarity of pieces of texts.

A more detailed comparison can be made by comparing the strings of checked texts. Most famous were these algorithms compare strings as suffix trees, naive string-search algorithm, Rabin–Karp algorithm, Knuth–Morris–Pratt algorithm, Boyer–Moore string-search algorithm, Bitap algorithm, Two-way string-matching algorithm, Backward Non-Deterministic DAWG Matching, Backward Oracle Matching, Aho–Corasick string matching algorithm, Commentz-Walter algorithm etc. [4]. Unfortunately these algorithms become ineffective at large length of strings.

A number of methods consists in breaking texts into parts (substrings) and comparing substrings of various documents among themselves. One of the most popular methods is the method of shingles, when chains of several words (usually 5) are concatenated into a string for which a hash code is calculated [5]. The resulting set of hash codes is a description of the content of the document. Comparing the sets of hash codes for different documents allows to evaluate the measure of their proximity – the greater the number of hash codes match, the more documents are similar to each other. The main disadvantage of the method is the direct dependence of the number of comparison of groups of hash codes on their number. For example, if the first document is represented by m hash codes, and the second is n, then the number of comparisons will be in the range from m to m×n. Considering that even small documents include thousands of words, the number of comparison operations for two documents will amount to millions [4].

Modern collections contain hundreds of thousands and sometimes millions of documents. This means that the number of operations to compare one document with a text collection will amount to trillions of operations! This is too much to ensure high speed. Despite the fact that there are approaches that allow reducing the number of comparison operations by several orders of magnitude (for example, considering only shingles that are dividing without a balance by some number), the comparison time still remains too large [6].

Another method is the n-gram method [7]. In the case of using n-grams with a size of several characters, the number of comparison operations will be significantly larger than in the method of shingles.

The second group of methods is signatures. The essence of signature methods is reduced to the presentation of a document by a certain code (hash function, number, checksum), which makes it possible to identify the same (or almost the same) documents with a high degree of probability. In fact, the comparison of documents is reduced to the comparison of several numbers – document signatures [8, 9]. This significantly reduces memory requirements and computational overhead. A distinctive feature of signatures is the ability to count them at any time, including in advance, and not at the time of verification.

Signatures can be built according to different principles. Most often, the content of the document is taken as a basis (single words or interrelated strings of words). The following signatures can be distinguished [10]:

- Document checksum (CRC, MD5) [11].
- Signatures calculated from the set of the most frequent words (based on the formulas TF, TF * IDF, TF * RIDF) [10, 12, 13, 14].
- Signatures based on the longest or significant sentences [8].
- Signatures based on the I-Match function [15, 16].
• Signatures based on the shingle method and its modifications (megashings, supershings) [5, 17].
• Signatures based on the vocabulary of description words [18].
• Rabin signature for counting fuzzy checksums of documents [19].
• Winnowing signature for fingerprinting documents [20] etc.

Due to the fact that most signatures describe the content of documents, changing the text can significantly reduce the possibility of their use, since the probability that the changes will affect the value of the signature is quite large.

3. Combined method of detecting fuzzy duplicates
The considered methods give either low search completeness at low computational costs, or acceptable completeness at very large computational costs. For this reason, a different approach needs to be developed. We tried to combine signature methods with context analysis methods to get a better result. The essence of the combined method of detecting fuzzy duplicates is as follows: at the beginning of the whole set of documents with the help of signatures similar texts are searched and then, using context analysis methods, a detailed comparison of the texts found in this way is carried out.

Let’s see which signatures are better for detecting fuzzy duplicates. Our previous studies showed the following results [21]. Signatures calculated from the set of the most frequency words (TF, TF * IDF, TF * RIDF) give a fairly balanced ratio of completeness and accuracy of the search. The signature based on the combination of the word frequency TF and the residual inverse frequency of the RIDF documents showed the best result. The disadvantage of this group of signatures is the limited number of words to be counted (most often 6 words). This leads to the fact that documents devoted to the same subject matter may have the same signature and erroneously be recognized as fuzzy duplicates. In addition, even minor changes to documents can lead to a rearrangement in the signature of words with a close arrangement of frequencies, which will lead to an erroneous recognition of documents as different.

Signatures, based on chains of the longest or most significant sentences (in terms of the weight of the words included in them according to the TD * IDF equation) show good precision, but with changes and compilations of texts often do not allow detecting similar documents.

The signature, built on the basis of the dictionary of description words, showed the most balanced meaning of precision and recall (as evidenced by the greatest value of the F-measure). In essence, it is similar to the signatures of the most frequent words, but it considers a substantially larger set of several thousand words. This is achieved by eliminating the influence of frequency changes on the quality of detection of duplicates.

In this way, the studies have shown that long and fuzzy signatures cope best with the task of fuzzy duplicate detections. A good candidate is signatures based on the vocabulary of description words. To calculate it several thousand words are selected from the entire dictionary built on the collection that describe the documents with the highest quality. For each document, a binary vector is calculated, the length of which is equal to the number of reference words. For each description word, the TF value is calculated [18]. If the value exceeds a certain threshold, 1 is written to the corresponding position of the vector, otherwise 0. The document signature is the specified binary vector.

As a measure of similarity, the number of coinciding nonzero values in the document vectors can be used, correlated to the total number of nonzero elements in these vectors. A measure of the proximity of two documents can be represented as:

\[ w(d_1, d_2) = 2 \left( \frac{|t^1_i \in d_1, t^2_i \in d_2 | t^1_i = 1, t^2_i = 1|}{|t^1_i \in d_1 | t^1_i = 1| + |t^2_i \in d_2 | t^2_i = 1|} \right) \]

where \(d_1, d_2\) – binary vectors for documents 1 and 2.
The value of the i-th element of the binary vector for documents 1 and 2, respectively.

The closer the measure of proximity \( w(d_1, d_2) \) is to one, the more documents are similar to each other.

To search for similar texts, just set a low threshold for proximity measures. Through experiments, a threshold of 0.5 was adopted for the proposed combined method.

![Figure 1. Combined method of detecting fuzzy duplicates.](image)

The shingle method is well suited for a detailed analysis of texts. Let's consider the combined method of detecting fuzzy duplicates in more detail. A long signature and a set of shingles are calculated for the analyzed document. The next step compares the received signature with the signatures of the collection of documents. This forms a list of documents with similar content. In accordance with this list, a set of shingles is calculated for found documents. The next step compares sets of shingles for documents. A measure of the similarity of documents is calculated as the proportion of matching shingles.

### 4. Research results

To test of combined method of detecting fuzzy duplicates the database of abstracts from several sources have been collected:

- Studentbank.ru site – 71193 documents.
- The site bestreferat.ru – 360534 documents.

A total of 431,727 documents were collected with a total text size of 30 GB. It should be noted that during the study only the text content of the documents was considered. Therefore, for documents in different formats with the same content, the signature values were the same.

In order to check two documents for all signature methods, it is enough to perform one comparison operation (table 1). For TF*IDF, the number of comparisons is equal to the size of the dictionary (in our case, more than 147 thousand terms).
For the shingle method, the number of shingles is equal to the number of words in the text minus the size of the shingle (5 words) plus 1 (on average, about 3800 shingles per document). Then the number of comparison operations will be equal to the product of the number of shingles in the checked documents (in our case, about 14 million operations).

For n-gram method, the number of n-grams is equal to the number of characters in the text, divided by the length of the n-gram (on average, about 12500 n-grams per document). Then the number of comparison operations will be equal to the product of the number of n-grams in the checked documents (in our case, about 156 million operations).

### Table 1. Number of comparison operations for two documents.

| Method                        | Number of comparison operations |
|-------------------------------|---------------------------------|
|                               | Equation                        | Test collection               |
| CRC                           | 1                               | 1                              |
| TF signature                  | 1                               | 1                              |
| TF*IDF signature              | 1                               | 1                              |
| TF*RIDF signature             | 1                               | 1                              |
| Long string signature         | 1                               | 1                              |
| Heavy string signature        | 1                               | 1                              |
| I-Match signature             | 1                               | 1                              |
| SuperShingles signature       | 1                               | 1                              |
| MegaShingles signature        | 1                               | 1                              |
| I-Match big signature         | 1                               | 1                              |
| Description words signature   | 1                               | 1                              |
| TF*IDF                        | Dictionary size                 | 147 400                       |
|                               | (Word count 1 – shingle size + 1)× | 14 440 000                    |
| Shingles                      | (Word count 2 – shingle size + 1) |                               |
|                               | (Char count 1 / n-gram size) ×  | 156 250 000                   |
|                               | (Char count 2 / n-gram size)    |                                |
| N-gram                        |                                  |                                |
|                               |                                  |                                |

### Table 2. Total number of comparison operations for collection.

| Method                        | Total number of comparison operations |
|-------------------------------|---------------------------------------|
| CRC                           | 431 727                               |
| TF signature                  | 431 727                               |
| TF*IDF signature              | 431 727                               |
| TF*RIDF signature             | 431 727                               |
| Long string signature         | 431 727                               |
| Heavy string signature        | 431 727                               |
| I-Match signature             | 431 727                               |
| SuperShingles signature       | 431 727                               |
| MegaShingles signature        | 431 727                               |
| I-Match big signature         | 431 727                               |
| Description words signature   | 431 727                               |
| TF*IDF                        | 63 636 559 800                        |
| Shingles                      | 6 234 137 880 000                     |
| N-gram                        | 67 457 343 750 000                    |
We analyzed the number of comparisons of one document with documents of the entire collection (table 2). For all signature methods, the number of comparison operations is equal to the number of documents (431,727). For TF*IDF, the number of comparisons is 63 billion.

For the shingle method, the number of comparison operations is 6 trillion, and for n-gram method – 67 trillion operations. The combined method showed a good result: the number of comparison operations is 72 million. It's almost 85,000 times less than for the shingle method and 800 times less than for the TF*IDF.

5. Conclusion
Signature methods allows quickly search for similar documents, but they are very sensitive to text changes. Using popular methods such as n-grams and shingles for large amounts of data leads to significant computational costs [13, 20]. Thus, the number of comparison operations for two documents will be in the millions. Due to the fact that modern collections contain hundreds of thousands, and sometimes millions of documents, the number of operations for comparing one document with a text collection will amount to trillions of operations! This is too much to ensure high speed.

Proposed combined method can significantly reduce the number of comparison operations and at the same time maintain the main advantages (precision and recall) of the shingles method. The method first performs an approximate search for similar documents using a long signature (description words signature) with a small match threshold. In the future, a detailed search for matches in previously found documents is performed using the shingles method.

Acknowledgments
The reported study was funded by RFBR according to the research project № 19-07-00692.

References
[1] Sharapova E 2014 One way to fuzzy duplicates detection Proc. of 14 International multidisciplinary scientific Geoconference SGM2014. Informatics, Geoinformatics and Remote Sensing. Conference proceedings (Albena, Bulgaria) I 273–7
[2] Salton G, Wong A and Yang C S 1975 A vector space model for automatic indexing Communications of the ACM 18 613–20
[3] Van Rijsbergen C J 1979 Information Retrieval (London: Butterworths)
[4] Gusfield D 1997 Algorithms on Strings, Trees and Sequences (Cambridge: Cambridge University Press)
[5] Broder A, Glassman S, Manasse M and Zweig G 1997 Syntactic clustering of the web Computer Networks and ISDN Systems 29 1157–66
[6] Sharapov R and Sharapova E 2018 The problem of fuzzy duplicate detection of large texts CEUR Workshop Proceedings 2212 270–7
[7] Sidorov G, Velazquez F, Stamatatos E, Gelbukh A and Chanona-Hernández L 2012 Syntactic Dependency-based n-grams as Classification Features Lecture Notes in Artificial Intelligence 7630 1–11
[8] Heintze N 1996 Scalable Document Fingerprinting Proc. of Second USENIX Workshop on Electronic Commerce (Oakland, USA) pp 191–200
[9] Manber U 1994 Finding similar files in a large file system Proc. of the Winter 1994 USENIX Conference (San Francisco, USA) pp 1–10
[10] Zelenkov Y and Segalovich I 2007 Comparative analysis of methods for fuzzy duplicate detection for Web-documents Proc. of 9-th Russian Scientific Conference «Digital Libraries: Advanced Methods and Technologies, Digital Collections» RCDL2007 (Pereslavl, Russia) pp 166–74
[11] Berson T A 1992 Differential Cryptanalysis Mod with Applications to MD5, EUROCRYPT'92
\textit{Lecture Notes in Computer Science} \textbf{658} 71–80

[12] Salton G and McGill M J 1983 \textit{Introduction to modern information retrieval} (McGraw-Hill)

[13] Salton G and Buckley C 1988 Term-weighting approaches in automatic text retrieval \textit{Information Processing & Management} \textbf{24} 513–23

[14] Church K and Gale W 1995 Poisson mixtures \textit{Natural Language Engineering} \textbf{1} 163–90

[15] Chowdhury A, Frieder O, Grossman D and McCabe M 2002 Collection statistics for fast duplicate document detection \textit{ACM Transactions on Information Systems} \textbf{20} 171–91

[16] Kolcz A, Chowdhury A and Alspector J 2004 Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization \textit{Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining} pp 605–10

[17] Fetterly D, Manasse M and Najor M 2003 A Large-Scale Study of the Evolution of Web Pages \textit{ACM} pp 669–78

[18] Ilyinsky S, Kuzmin M, Melkov A and Segalovich I 2002 An efficient method to detect duplicates of Web documents with the use of inverted index \textit{Proc. of the 11th International World Wide Web Conference} (Honolulu, USA)

[19] Rabin M 1978 Digitalized signature as intractable as factorization \textit{Technical Report MIT/LCS/TR212} (MIT Laboratory for Computer Science)

[20] Schleimer S, Wilkerson D S and Aiken A 2003 Winnowing: Local Algorithms for Document Fingerprinting \textit{Proc. of ACM SIGMOD International Conference on Management of Data} (San Diego, USA)

[21] Sharapova E and Sharapov R 2019 Detection of Fuzzy Duplicate Texts in News Feeds \textit{Proc. of Systems of Signal Synchronization, Generating and Processing in Telecommunications} (Yaroslavl, Russia) pp 1–5