Identification of Key Sentences in the Task of Text Duplicate Detection

E. V. Sharapova
Murom Institute, Vladimir State University
Murom, Russia
mivlgu@mail.ru

***Abstract*— The paper considers the problems of detecting duplicates of large text documents. To reduce the verification time, it is proposed to submit the document being verified with a set of key sentences. Key sentences are selected for parts of the text and are used to search for matches over the Internet. As a criterion for choosing key sentences, the largest sum of the weights of the words included in the sentence is calculated, taking into account the global frequency of words. Studies have shown that the use of key offers can significantly reduce the number of queries to search engines. At the same time, good duplicate text detection results are preserved.**

Keywords—key sentence; duplicate; text; sentence; search engine

I. Introduction

The problem of identifying duplication of documents becomes relevant due to the large amount of information available on the Internet. Many copies of the same documents can be stored on the Internet [1, 2]. At the same time, documents can be presented in whole, in separate pages or parts. In addition to determining duplicates, the problem arises of determining the originality of texts. For these purposes, various systems are being developed [3, 4, 5]. For small texts, the task is successfully solved. But for texts of large sizes, a number of problems arise. Search time is significantly increased, or duplicate detection quality becomes low.

II. The Problem of Detecting Duplicates of Large Texts

To search for duplicate documents on the Internet, need to compare the contents of the analyzed document with all documents on the Internet. Since the amount of information on the Internet is huge, it is impossible to download and analyze all available pages (documents) for relatively short periods. To accelerate the analysis, the content of the sites is downloaded to local drives and indexed. But the number of sites is huge. There are currently over 1.7 billion sites. Hundreds of terabytes will be required to store so much information. It takes a huge amount of time to read so much information from a hard drive. For this reason, most often the process of searching for duplicates is carried out using ready-made solutions - search engines. Search engines already have their own search indexes, optimized for quick work with huge amounts of information.

Duplicate search systems send queries to search engines in the form of checked text parts and analyze the results obtained from them [6]. Ideally, the request should be the full text of the document being analyzed. But search engines (Google, Yandex) limit the query size to a few dozen words. For example, Yandex imposes the following restrictions on requests: the maximum request length is 400 characters, the maximum number of words is 40. For this reason, duplicate search systems must break down the checked document into small parts and search for each of them. But here there are several problems. Firstly, for large documents the number of parts will be quite large, and their verification also requires a lot of time. Secondly, search engines limit the number of requests processed from users. In other words, they do not allow the formation of a large number of requests from users throughout the day.

Because of this, an important task is to reduce the number of queries to search engines while maintaining an acceptable quality of duplicate searches. This problem does not have a single solution and is an optimization.

What are the features of the check:

1. Uniformity. Verification should be carried out uniformly throughout the text. It is not enough to select several key phrases from the beginning or middle of the text being checked.

2. Adaptability to the size of the text. You can’t just break all the text into pieces of 400 characters and sequentially send requests to search engines. This works well on small texts. But for long texts, the number of such pieces can be very large and search engines will simply stop processing requests.

3. Speed of check. The waiting time for the results of checking any texts of course size - the user can wait several minutes, hours. But he will not wait for hundreds of hours, tens of days.

4. Quality check. It is not enough to take one document for each request. You need to check a lot of documents (from 3 to at least 10) for each request to search engines.

III. Presentation of a Document by a Set of Key Sentences

One of the main problems in finding duplicate texts on the Internet is the large number of queries to search engines [7]. The number of queries is directly proportional to the size of

978-1-7281-4944-8/20/$31.00 ©2020 IEEE

the document. When solving a problem head on, the number of queries is equal to the number of sentences in the document.

Naturally, the problem arises of reducing the number of queries to an acceptable value of N. One of the solutions to the problem is to present the document with a set of key sentences [8]. Each of these sentences will serve as a separate query to the search engine. Thus, by choosing N key sentences, we can reduce the number of queries to search engines to N.

The main objective of the key sentences is to serve as the best representation of the contents of the document. In other words, the use of a key sentence as a request should provide a selection of search documents that are closest in content to documents. The key sentence should be the most original. It should not consist of the most used phrases, such as “Today is a beautiful morning” or “Good afternoon, dear colleagues”. If the sentence is too widespread, then the result of the search engine query will be a large list of documents found, among which a really similar document can be skipped. The fact is that when searching for duplicate texts on the Internet, the first few search results (for example, 10 links from the first page) are usually chosen to reduce the time spent on work. Accordingly, if the desired document is not located on the first page of search results, then most likely it will not be considered by the verification system. It follows that the key sentence should be, firstly, long enough, and secondly, the most unique, to ensure that duplicates (if any) are in the first lines of search results.

We propose the following approach to identify key sentences:

1. The entire document is divided into N parts [9].

$$d = \{p_1, p_2, p_3, ..., p_N\}$$

where p_i – i-part of the document.

Parts can be formed page by page (for example, one or two pages), for a certain number of characters (for example, 2000 characters), for a certain number of sentences (for example, 10 sentences), or by dividing a document into a specified number of parts. At the same time, the boundaries of the parts are set along the boundaries of the sentences (sentences are not torn into parts).

2. A list of its sentences is formed for each part:

$$p_i = \{s_1^i, s_2^i, s_3^i, ..., s_m^i\}$$

where s_k^i – k-th sentence in the i-th part of document.

3. In each part, according to a certain algorithm, the most significant (key) sentence is revealed:

$$ks_i = \max f(s_1^i, s_2^i, s_3^i, ..., s_m^i)$$

where $f(s_k^i)$ – a function of calculating the weight of the k-th sentence.

4. A set of key sentences is being formed:

$$KS = \{ks_1, ks_2, ks_3, ..., ks_N\}$$

Key sentences from the set in the form of queries are sent to search engines to search for similar documents on the Internet.

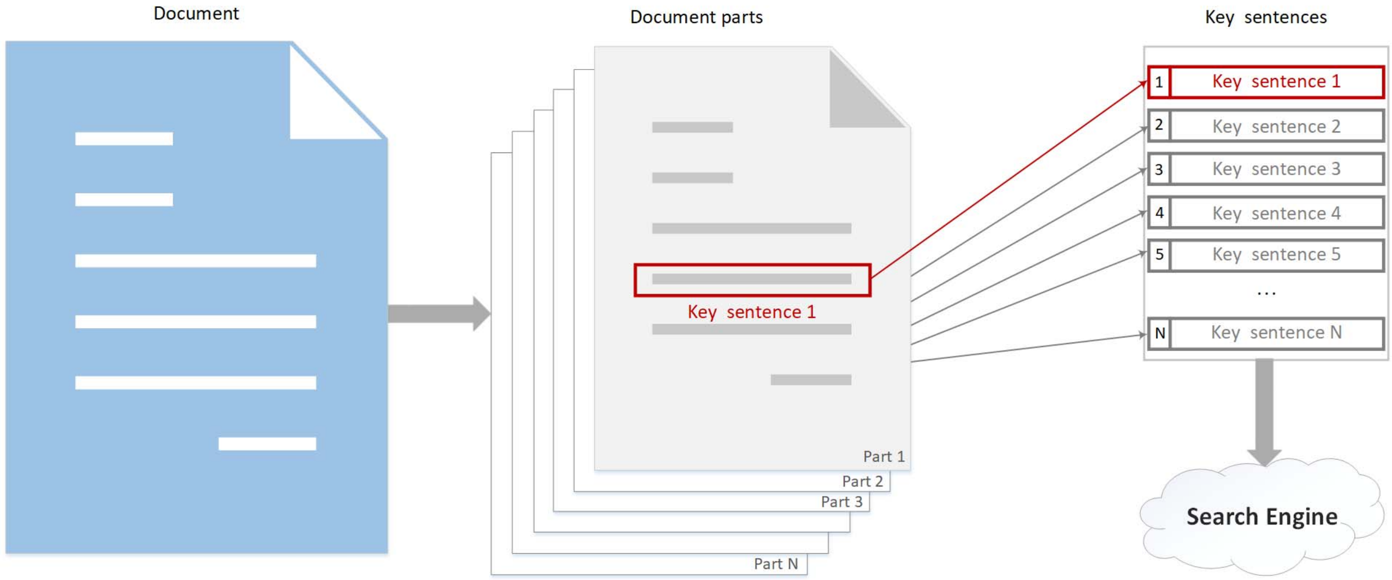

Fig. 1. Presentation of a document by a set of key sentences

IV. Methods of Key Sentences Identification

The key sentences can be selected according to various criteria. Of them can be identified [10, 11]:

- The longest sentences,
- Sentences with the highest sum of word weights,
- Sentences with the highest average weight of words,
- Sentences with the most count of significant words.

The weight of words can be determined in different ways. Often used TF*IDF method (term frequency–inverse document frequency) in various modifications [12, 13, 14, 15].

Term frequency (TF) is frequency of word occurrence in a document.

$$tf(t) = \frac{n_t}{\sum_k n_k}$$

where n_t – the number of occurrences of the word t in the document,

$\sum_k n_k$ – total number of words in the document.

Inverse document frequency (IDF) is total occurrence of words in all documents in the collection [16].

$$idf(t) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$$

where |D| – number of documents in the collection,

$|\{d_i \in D | t \in d_i\}|$ – number of documents in which the word t occurs.

$$tf\text{-}idf\,(t) = tf\,(t) \times idf\,(t)$$

The inverse document frequency IDF can be calculated in collection of documents, single document or one part of a document [17].

In the task of choosing key sentences, each sentence is actually a separate unit. Since the selection of key sentences is carried out in each part separately, the words are only influenced by the choice from the part in question. Accordingly, it is sufficient to use only the word frequencies in each part of the document. The disadvantage of this approach may be the overestimated weight of commonly used words, especially in small parts of the text. A situation is possible when all words in a part will occur once, and, accordingly, have the same weight.

On the other hand, all parts are one document, that is, between them there is a semantic connection. For this reason, the occurrence of words in other parts may indirectly indicate the importance of the word in the considered part. Then, counting word frequencies throughout the document can also be justified.

Accounting for the occurrence of words throughout the collection also makes sense. This allow identifying high frequency, commonly used words. Such words in small texts can get unjustifiably high (overestimated) weight. For this reason, it makes sense either to take into account the occurrence of words in the entire collection of documents, or to filter high-frequency words (for example, based on Zapf's law [18] or available statistics).

V. Quality Assessment of Key Sentences

To assess the impact of the selected key sentences on the quality of duplicates searching on the Internet, a study was conducted, the essence of which was to evaluate the search results [19].

The higher the duplicate found is in the search results (ideally in the first place), the better the key sentence represents analyzed text.

It should be noted that the documents found by search engines will almost always correspond to the search query. The problem is that the documents found may correspond to the query, but not be duplicates of the analyzed document. In other words, key sentences can be found in other documents that are little related in content to the analyzed one. Therefore, the more unique the sentence, the more likely it is that the documents found by search engines will indeed be duplicates.

For evaluation, we used the metric "Response Value" (ReciprocalRank), which allow to evaluate how much effort it takes for the program to find the first answer to query, or what is the likelihood that the program will search the results to the position where the first correct answer is [20, 21, 22]. Formally, the "value" of an answer to a specific task is calculated as:

$$ReciprocalRank = rank(pos),$$

where *pos* is – minimum position that the relevant response is at.

If there are no correct answers in the answer, then the "value" is 0.

The *rank(pos)* function is usually defined by a certain ruler of values for the first few positions and is considered equal to 0 for all others.

Function

$$rank(pos) = 1\,/\,pos$$

ReciprocalRank is inversely proportional to the position of the first relevant system response.

For testing, several options for the formation of key sentences were selected.

To evaluate the effect of long sequences of words in queries, sentences with the longest length were selected. To assess the influence of word weights calculated using different methods, we selected sentences with IDF counting for the analyzed part of the document, the entire document and the global collection of documents. Sentences with the largest number of most significant words on a selection basis are similar to the longest sentences, but in this case, commonly used, high-frequency words are not taken into account. Due to this, the sequence of words in the selected sentences is more rare, and, as a result, more unique.

The sentences with the highest average word weight are keyword-rich sentences. Moreover, the length of such sentences is not always large, and may be only a few words, which is not always enough.

To compare the quality of the selection of key sentences, the study added options for generating queries from random sentences of the analyzed text.

In contrast to long and heavy sentences, short sentences, sentences with the least weight of the words included in them, were also chosen as a comparison.

For the study, 10 documents were selected from different collections of essays, each with a volume of 30-50 pages. Documents were divided into parts of 2000 characters each. The test results are shown in table I.

TABLE I. TEST RESULTS

Key Sentence	ReciprocalRank
Short sentence	0.23
Most "light" sentence	0.26
Random sentence	0.64
Long sentence	0.81
The most "heavy" sentence (part)	0.62
The most "heavy" sentence (document)	0.77
The most "heavy" sentence (global)	0.84
Sentence with highest average weight of words	0.58
Sentence with the most significant words	0.82

As can see, short sentences and the most “lightest” sentences give a very small ReciprocalRank value. This means that for such queries there is a large number of documents that contain these sentences, but are not duplicates of the source text. Both types tend to short sentences with common words.

The best result is given by the most “heavy” sentences (global) and sentences with the most significant words. The long sentences are inferior to them. All three types tend to the longest sentences containing important, low-frequency words.

VI. CONCLUSION

The approaches proposed in this work to find duplicate documents using a set of key sentences are used in the Author.NET system. As a criterion for choosing key sentences, the largest sum of the weights of the words included in the sentence is calculated, taking into account the global frequency of words.

Studies have shown that using key sentences can reduce the number of queries to search engines by ten times. At the same time, good duplicate text detection results are retained.

A further development of the work may be the optimization of the selection of the parts size into which the analyzed text is divided. In addition, it is of interest to analyze the proximity of the content of key sentences when choosing them. This can reduce the number of key sentences or make them more diverse.

Acknowledgment

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

References

[1] R. Sharapov and E. Sharapova, “The problem of fuzzy duplicate detection of large texts,” CEUR Workshop Proceedings, vol. 2212, pp. 270-277, 2018.

[2] J. Dean and M. Henzinger, “Finding related pages in the World Wide Web,” Computer Networks, vol. 31, pp. 1467-1479, 1999.

[3] S. Brin, J. Davis, and H. Garcia-Molina, “Copy detection mechanisms for digital documents,” Proc. ACM SIGMOD Annual Conference, pp. 398-409, 1995.

[4] M. R. Henzinger, “Finding near-duplicate web pages: a large-scale evaluation of algorithms,” SIGIR 2006, pp. 284-291, 2006.

[5] C. Hoad and J. Zobel, “Methods for identifying versioned and plagiarised documents,” Journal of the American Society for Information Science and Technology, vol. 54, pp. 203-215, 2003.

[6] E. Sharapova, “One way to fuzzy duplicates detection,” Proc. of 14 International multidisciplinary scientific Geoconference SGEM2014. Informatics, Geoinformatics and Remote Sensing. Conference proceedings, vol. 1, pp. 273-277, 2014.

[7] E. Sharapova and R. Sharapov, “System of fuzzy duplicates detection,” Applied Mechanics and Materials, vol. 490-491, pp. 1503-1507, 2014.

[8] D. Gusfield, Algorithms on Strings, Trees and Sequences. Cambridge University Press, 1997, 556 p.

[9] D. Fetterly. M. Manasse and M. Najor, “A Large-Scale Study of the Evolution of Web Pages,” ACM, pp. 669-678, 2003.

[10] Y. Zelenkov and I. Segalovich, “Comparative analysis of methods for fuzzy duplicate detection for Web-documents,” Proceeding of 9-th Russian Scientific Conference «Digital Libraries: Advanced Methods and Technologies, Digital Collections» RCDL2007, pp. 166-174, 2007.

[11] J. W. Cooper, A. R. Coden, and E. W. Brown, “Detecting similar documents using salient terms,” Proc. 1st International Conf. on Information and Knowledge Management CIKM 2002, pp. 245-251, 2002.

[12] G. Salton and M.J. McGill, Introduction to modern information retrieval. McGraw-Hill, 1983. 448 p.

[13] G. Salton, A. Wong and C.S. Yang, “A vector space model for automatic indexing,” Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.

[14] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513-523, 1988.

[15] H.C. Wu, R.W.P. Luk, K.F. Wong and K.L. Kwok, “Interpreting TF-IDF term weights as making relevance decisions,” ACM Transactions on Information Systems, vol. 26 (3), 2008.

[16] S. Robertson, “Understanding inverse document frequency: On theoretical arguments for IDF,” Journal of Documentation, vol. 60 (5), pp. 503-520, 2004.

[17] C.D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[18] G.K. Zipf, Human Behavior and the Principle of Least Effort. Addison-Wesley Press, 1949. 573 p.

[19] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe, “Collection statistics for fast duplicate document detection,” ACM Transactions on Information Systems, vol. 20, pp. 171–191, 2002.

[20] M. Ageev, I. Kuralenok and I. Nekrestyanov, “Official ROMIP 2010 metrics,” Russian Workshop on Evaluating Information Search Methods. Proceedings of ROMIP 2010, pp.172-187, 2010.

[21] E.M. Voorhees, “Proceedings of the 8th Text Retrieval Conference,” TREC-8 Question Answering Track Report, pp. 77–82, 1999.

[22] O. Chapelle, D. Metlzer, Y. Zhang and P. Grinspan, “Expected reciprocal rank for graded relevance,” Proceeding of the 18th ACM Conference on information and Knowledge Management CIKM '09, pp. 621-630, 2009.