Large-scale Chinese Text Infringement Detection Based on Dual-Semantic Fingerprinting

Ruixue Zhao[1,a], Xiao Yang[2,a], Honglei Li[3,c*]

[1]Key Laboratory of Knowledge Mining and Knowledge Services in Agricultural Converging Publishing of National Press and Publication Administration; Agricultural Information Institute, Chinese Academy of Agricultural Sciences. Beijing, China.
[2]Key Laboratory of Knowledge Mining and Knowledge Services in Agricultural Converging Publishing of National Press and Publication Administration; Agricultural Information Institute, Chinese Academy of Agricultural Sciences. Beijing, China.
[3*]School of Management, Liaoning Normal University. Dalian, China.

[a]zhaoruixue@caas.cn;[b]736834185@qq.com
[c*]lhl@lnnu.edu.cn

Abstract. The SimHash algorithm is a type of hash method used to deduplicate large web pages. It is also widely used in text similarity comparison due to its high effectiveness and efficiency. In this paper, we improve the classical SimHash algorithm in semantic similarity detection of large Chinese texts. In our method, word similarity is first calculated using the text similarity determination method based on CiLin path depth algorithm, then the keywords extracted using TF-IDF are processed for synonym redundancy. Finally, dual-semantic fingerprints are generated and the Hamming distance between the fingerprints is calculated. The experimental results show that this improved SimHash algorithm is superior to the classical SimHash algorithm in terms of F1_score. It is suggested that this algorithm can further improve the probability of semantically finding infringing texts and provide technical support for digital copyright infringement detection.

Keywords: Infringement detection; Text similarity; CiLin; Dual-semantic fingerprinting

1 INTRODUCTION

Digital Rights Management (DRM) is the primary means of copyright protection for digital works distributed over the Internet. DRM is defined by the Association of American Publishers as the technology, tools and processes that protect intellectual property rights during digital content transactions. DRM is considered a systematic solution, including information security technology, to ensure the normal use of digital information (such as digital images, audio, video, etc.) by legitimate and authorised users, while protecting the copyright of the creators and owners of digital information, generating legitimate revenue based on copyright information, and identifying the copyright of

B. K. Kandel et al. (eds.), *Proceedings of the 2023 3rd International Conference on Business Administration and Data Science (BADS 2023)*, Atlantis Highlights in Computer Sciences 19,
https://doi.org/10.2991/978-94-6463-326-9_25

digital information in the event of copyright infringement. In DRM, digital copyright protection technology is a set of software and hardware technologies that protect the intellectual property rights of various digital content, ensure the legal use of digital content throughout its life cycle, balance the interests and needs of various stakeholders in the digital content value chain, and promote the overall digital market and information dissemination. Digital copyright protection technology covers the entire process of digital content circulation, from production to distribution, from sale to use, and involves the entire digital content value chain.

The Internet has generated a large amount of digital content and has greatly facilitated the sharing and distribution of digital content. However, it has also exacerbated the problem of digital copyright infringement, with text infringement becoming more serious [1]. Digital content infringement detection is an important work in copyright protection of digital works, and its technical basis is content similarity comparison. Therefore, the use of text similarity algorithms to quickly find possible similar texts from large amounts of text and to assist in determining infringements will undoubtedly greatly improve the efficiency of infringement detection.

The SimHash algorithm and the similarity calculation method based on CiLin are both mainstream text similarity algorithms. In this paper, we propose a large-scale Chinese text similarity measure based on dual-semantic fingerprinting and apply it to text infringement detection in the field of copyright protection. The method combines the improved SimHash algorithm with the lexical similarity calculation of CiLin to generate the dual-semantic fingerprints of texts by synonymous substitutions, and calculates them separately. This not only accounts for text semantics and possible malicious manipulation, but also addresses the need for efficient similarity detection for large texts.

2 RELATED STUDIES

Currently, many text similarity methods have been proposed. Wang et al. classified them into surface text similarity (STS) and semantic similarity (SS) [2]. STS directly targets the original texts and acts on string sequences or character combinations, using the degree of character correspondence or the distance between two texts as the measure of similarity. It is usually divided into character-based and term-based, according to the methods of computational granularity. In response to the problem that STS only processes superficial words regardless of semantics, SS is proposed. SS mainly includes knowledge-based and corpus-based hybrid methods to improve the effectiveness.

The surface text similarity calculation methods represented by Vector Space Model (VSM) and Bag-of-Words (BOW), which take the text as a vector for calculation. The feature vectors generated by the above methods are usually high-dimensional, sparse and lack semantics. Peng et al. proposed a Chinese text similarity calculation method based on concept similarity. In the method, the text is transformed into a lexical vector space model, and the lexicon is divided into a set of concepts [3]. The similarity between words is obtained by calculating the inner product between concepts, and finally the text similarity is calculated based on the word similarity. However, for large text,

two types of algorithms based on set and vector space models have problems of high workload, low efficiency and poor accuracy [2].

It is well known that hash algorithm-based detection techniques are characterised by unidirectionality, collision resistance and uniformity of mapping distribution, which have been shown to significantly reduce memory computation overhead and detection time by mapping text into unique fixed-length binary codes. Li et al. proposed a repeated comment detection method based on cryptographic hash matching techniques (e.g. SHA-1 and MD5) and used it to determine the similarity of forum comments. In their experiments, the proposed method showed better performance than classical algorithms such as I-Match and DSC [4]. Since classical hash algorithms are based on randomly mapping original texts into unique hash values, they can determine whether two texts are the same, but cannot measure the similarity between them [5]. In addition, the technical characteristics of collision resistance and avalanche effect will cause violent disturbance in the hash values, even slight changes occur in the original texts. This means that hash values generated by classical algorithms cannot fairly and accurately measure similarity.

To address the problems associated with classical hashing methods, Charikar proposed the SimHash fingerprinting technique to make similar texts produce similar hashes, thus laying a theoretical foundation for achieving slightly modified digital fingerprint similarity detection [5]. Manku et al. demonstrated Charikar's fingerprinting technique on a multi-billion repository of web documents, solving the Hamming distance problem for fast search of similar fingerprints [6]. Currently, SimHash has been widely used in text deduplication [7-9], code clone detection [10-11], approximate text retrieval [12-17], encrypted data search [18-19], document similarity detection [20-24], etc. However, the classical SimHash algorithm can only achieve surface text similarity calculation and cannot compare the deep rich semantics of Chinese text, which should be improved in semantic similarity calculation.

3 METHODS

Even though the SimHash algorithm has been shown to be effective for near-duplicate detection of large web pages and texts [6-9], the generated fingerprints can only achieve a superficial text similarity comparison if the semantics of the text are ignored, which results in low effectiveness of recognition of synonyms and polysemous words. It is necessary to consider the semantics to further determine the similarity between large texts. Since that the CiLin-based similarity measure algorithm has achieved better results in experiments compared to manual determination [25], which can redundantly process the keywords extracted from the text with synonyms, this paper combines them to propose a dual-semantic fingerprinting approach. Fig. 1 shows the flow of the proposed algorithm.

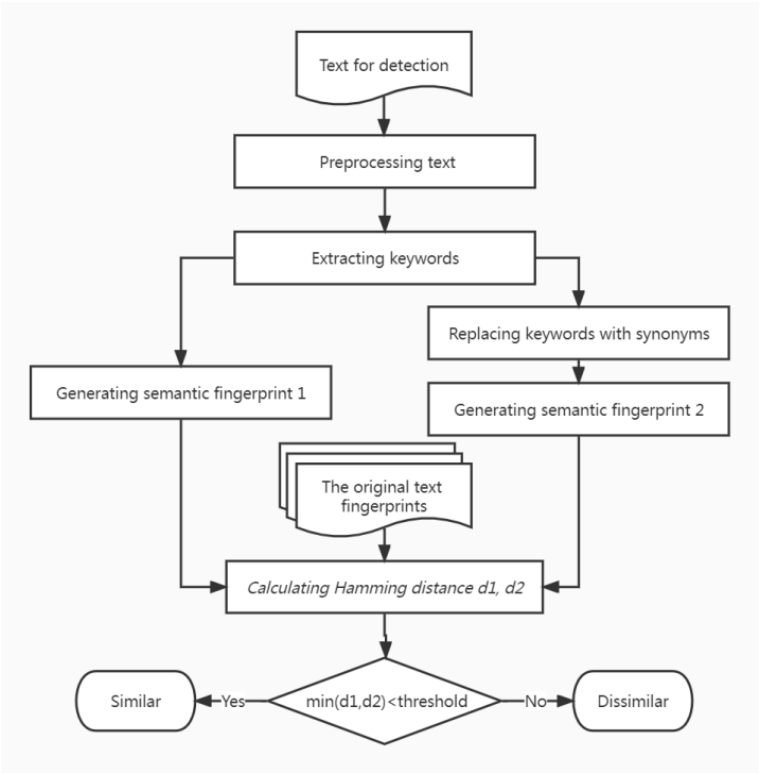

Fig. 1. Flowchart of dual-semantic fingerprinting-based text similarity detection

3.1 Pre-processing Texts

In the field of natural language processing, a text is usually considered as a collection of sentences, which are regarded as sequences of words, phrases. Since texts are written in various formats, each text is usually converted to a plain text for ease of processing. In practice, tokenisation is used to pre-process the texts. Text pre-processing usually involves punctuation erasure, n-char(s) filter, stop-word-filter and case-converter [27]. For Chinese texts, word segmentation and part-of-speech filtering are conducted in advance. After the text has been converted into a set of words, TF-IDF [28] or TextRank [29] algorithms are often used for keyword extraction. Qian et al. found that keywords extracted using TextRank were significantly less effective than TF-IDF, which extracted keywords with more occurrences and ignored the importance of the words, while TF-IDF avoided these problems and ensured that the extracted keywords could reflect the main content of the text.

Since the SimHash algorithm generates hash values by segmenting words in the full text, a large number of non-keywords will contribute little to the semantic expression and reduce the accuracy of text fingerprinting, making the text fingerprints generated

with low efficiency. In this study, TF-IDF is used to extract keywords and generate text fingerprints of keywords instead of full text.

3.2 Generating text fingerprints

In this study, SimHash is used to generate text fingerprints. As a typical locality sensitive hash (LSH) algorithm, SimHash is used to perform fast approximate nearest neighbour search in high-dimensional data sets [5]. The core idea of SimHash is random projection. For two data points mapped to a low-dimensional dataset, if they are adjacent in the original set, it is highly likely that they are still adjacent in the new low-dimensional dataset, and vice versa. This suggests that there is a high probability that samples will remain similar to each other after dimension reduction. SimHash maps high-dimensional textual feature vectors into binary codes with a fixed number of bits (e.g. 64 bits). The binary codes are referred to in this paper as "fingerprints". Fingerprints are sensitive to a few bits, so the similarity between fingerprints can be measured by Hamming distance or Levenshtein distance.

The SimHash algorithm uses random projection as its hash function. Given a collection of vectors, SimHash defines the family of functions in Eq. (1).

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & if\ \vec{r} \cdot \vec{u} \geq 0 \\ 0 & if\ \vec{r} \cdot \vec{u} < 0 \end{cases} \tag{1}$$

Where $\vec{r} \cdot \vec{u}$ represents the dot product of the vectors $\vec{u}$ and $\vec{r}$. For vector $\vec{u}$ and $\vec{v}$, there is

$$Pr[h_{\vec{r}}(\vec{u}) = h_{\vec{r}}(\vec{v})] = 1 - \frac{\theta(\vec{u},\vec{v})}{\pi} \tag{2}$$

Eq. (2) illustrates the probability that their hash values are the same value at a given bit, where $\theta(\vec{u}, \vec{v})$ is the angle between the two vectors.

3.2.1 Generating the first text fingerprint.

We use TF-IDF to extract keywords from the text. For proper nouns in a domain that occur repeatedly in the text set, the IDF will reduce the importance of the text, which is the shortcoming of TF-IDF. Therefore, the weight calculation of keywords needs to be adjusted in conjunction with more semantic factors, such as word length. In practice, Wang et al. used the topic relevance of words as an additional weight and the length of the terminology vocabulary as a basis for judging the topic relevance of words. They obtained the fitted normal distribution function of word length by counting the lengths of 10,000 Chinese terms in the CSSCI keyword database and performing a normal fit in Eq. (3).

$$f(x) = 2510 \times e^{-\left(\frac{x-4.51}{2.207}\right)^2} \tag{3}$$

The Chinese term length function is defined in Eq. (4).

$$len(x) = 2510 \times \frac{e^{-\left(\frac{x-4.51}{2.207}\right)^2}}{10000} = 0.251 \times e^{-\left(\frac{x-4.51}{2.207}\right)^2} \quad (4)$$

Where x is the length of the word. Undoubtedly, the closer the word length is to 4.51, the higher the function value of the function. This means that the word has a higher thematic relevance. Combining the TF-IDF with the length function in Eq. (4), the weight of the keywords is calculated in Eq. (5).

$$w(i) = tf_{i,j} \times idf_i \times \left(1 + len(x_i)\right) \quad (5)$$

Where x_i is the length of word i.

As an improved weight function, $w(i)$ is used to generate the first semantic fingerprint of the text.

3.2.2 Generating the second text fingerprint.

It is well known that synonyms express the same meaning in different words, resulting in different text fingerprints. This means that a text can evade infringement detection by replacing words in the text with synonyms. In this study, the keywords of the compared texts are replaced with synonyms and fed into the SimHash programme to generate the second fingerprints. We use CiLin to obtain synonyms of keywords.

CiLin is a classical Chinese synonym dictionary compiled by Mei et al. in 1983 [30]. CiLin is computable. It was originally designed to classify and categorise Chinese synonyms and homonyms, and then extended by the Information Retrieval Laboratory of the Harbin Institute of Technology. In the extended version of CiLin, entries are organised in a five-level tree-like structure. In the tree-like category, nodes in each level belong to five categories: major category, middle category, subcategory, word group, and atomic word group.

In CiLin, the synonym is discovered by a path distance-based algorithm [2]. The algorithm computes the similarity between two words by the distance of their shortest path and the node depth in the ontology structure [25]. In this algorithm, for any two senses s_1 and s_2, their similarity is computed in Eq. (6).

$$Sim(s_1, s_2) = \frac{Depth(LCP(s_1,s_2))+\alpha}{Depth(LCP(s_1,s_2))+\alpha+Path(s_1,s_2)+\beta} \quad (6)$$

Where LCP is the nearest common parent node of the word senses s_1 and s_2. $Path_1$, $Path_2$ are the path distances from s_1 and s_2 to their nearest common parent nodes, respectively. $Depth$ is the depth distance from the nearest common parent node of s_1 and s_2 to the root node. $Path(s_1, s_2) = Path_1 + Path_2$ represents the shortest path between two senses. α is the depth adjustment parameter and β is the path adjustment parameter.

Considering that there are words with multiple senses, the final similarity of the two words is the one with the greatest similarity among all pairs of senses. Given that the word has senses, its similarity in CiLin is calculated in Eq. (7).

$$Sim(w_1, w_2) = \max_{1 \le i \le m, 1 \le j \le n} \left\{Sim\left(s_{1i}, s_{2j}\right)\right\} \quad (7)$$

Where $Sim(s_{1i}, s_{2j})$ is the similarity value of the i-th sense of word w_1 to the j-th sense of word w_2.

The Common Nouns dataset published by Miller & Charles (M & C) and its manual scores are used as the standard, which consists of 30 English noun pairs with high, medium and low semantic similarity, respectively [26]. Eq. (10) is used to calculate the similarity of this dataset and obtain its Pearson correlation coefficient with the M & C manual value. Then, using the word similarity calculation method based on CiLin in Reference [25], words with similarity to keywords equal to 1.0 are extracted for their synonymous substitution, and the second semantic fingerprint of the text is generated in the same way.

After the above process, each text will generate dual-semantic fingerprints, as shown below.

0010000111111000110101111011101001111010111110110001111101100100 (a)

0010000111101000101001110011101001001010100111110101111001100011 (b)

The first one (a) is output by the SimHash algorithm, and the other one (b) is output by the same algorithm after a synonym replacement operation.

3.3 Calculating text similarity

The Hamming distance is used to measure the similarity between the dual-semantic fingerprints of the text and the fingerprints stored in the fingerprint repository. The smaller the Hamming distance is, the more similar they are. Therefore, the minimum value of the two comparisons is taken as the final result.

For the 64-bit SimHash value, the Hamming distance $k = 3$ is generally taken as the similarity threshold [6], which can also be adjusted in practice. That is, if the final comparison value of the text to be detected is not higher than the threshold, it can be judged as an infringed text.

For two 64-bit binary fingerprints f_1 and f_2, the Hamming distance between them is defined in Eq. (8).

$$HammingDistance = \sum(f_1[i] \oplus f_2[i]) \tag{8}$$

Where $0 \leq i \leq 63$, $\oplus$ is the XOR operator and $f[i]$ is the i-th bit of the fingerprint code.

4 Experiments

4.1 Experimental Data

In this paper, the Chinese text classification corpus of Fudan University is used to test the proposed method. The corpus contains 9,833 documents belonging to twenty categories. Forty texts with more than 800 characters are randomly selected as the basic

data set, which are processed by the above methods to form the text fingerprint set for similarity calculation.

Back-translation, as a common data augmentation method in NLP, is used to generate sentences in a generative way without deviating from the original semantics. As a result, all texts in the basic data set are back-translated in the "Chinese-English-Chinese" way to generate an approximate text set. Twenty texts outside the data set are mixed as a test set.

4.2 Experiment design

In this study, the proposed method is coded in Python 3.6. Jieba is used for text segmentation and keyword extraction. F1_score is used to evaluate the proposed method. By comparing the F1_scores of the single-semantic fingerprinting method and the dual-semantic fingerprinting method on the same dataset, we validate the performance of the dual-semantic fingerprinting-based method proposed in this paper.

Since the Hamming distance threshold is critical in determining the results of the violation detection, different algorithms will perform differently with different thresholds. Therefore, we measured F1_scores for each of the three algorithms at different Hamming distance thresholds in hypothesis testing.

Table 1. Extracted keywords, substituted synonyms and their weights (Top 10)

Keyword*	Weight (TF-IDF)	Weight (TF-IDF + Length)	Synonym*	Weight (TF-IDF+ Length)	Same
Yishu	0.403	0.431	Zhuanao	0.431	N
Jiaoyu	0.264	0.283	Jiaoyang	0.283	N
Xuesheng	0.166	0.177	Xuetong	0.177	N
Xuexiao	0.070	0.075	Muxiao	0.075	N
Meide	0.066	0.070	Junde	0.070	Y
Huhuan	0.064	0.069	Ganzhao	0.069	N
Jiaowei	0.060	0.074	Jiaoyuju	0.074	Y
Bizhe	0.055	0.059	Qicaoren	0.064	N
Yinyue	0.055	0.058	Yinyue	0.058	Y
Zhongxue	0.049	0.053	Guoxue	0.053	N

*Since all words in the manuscript should in English, we replace all Chinese words in this table with Pinyin.

Table 1 shows the top 10 keyword examples with the highest weights in this text, and whether the substituted synonyms are the same as the original ones. From the table, we can see that the word weights have increased after incorporating the word length factor. Meanwhile, limited by the scope of CiLin inclusion and the way of similarity calculation, the synonyms of some words after substitution are consistent with themselves, so not all keywords can achieve synonym substitution. A total of 85 keywords were extracted from this text, and a total of 52 synonyms were substituted, achieving 61.2% synonym redundancy, and the substituted word weights also varied with word length.

4.3 Results and Discussions

Taking an educational text in the experimental data as an example, we observed the keywords extracted by TF-IDF and their weights, the changes in weights after adding the word length factor, as well as the synonyms and their weights after CiLin synonym substitution.

Fig. 2 shows the F1_score of the three methods at different Hamming distances, and it can be seen that dual semantic fingerprinting performs optimally at most thresholds. The F1_score of each method also grows with the threshold value, and the original SimHash algorithm is significantly weaker than the semantic fingerprinting method in terms of growth rate. In addition, both semantic fingerprinting algorithms have reached their maximum value at $k = 18$, while the classical SimHash algorithm reaches its optimum at $k = 25$. As the Hamming distance measures the degree of difference between fingerprints, once the threshold reaches a certain level, it can be assumed that all texts are similar and the F1_score will gradually converge, with the dual-semantic fingerprinting converging at k=24 and the other two methods both converging at k=26.

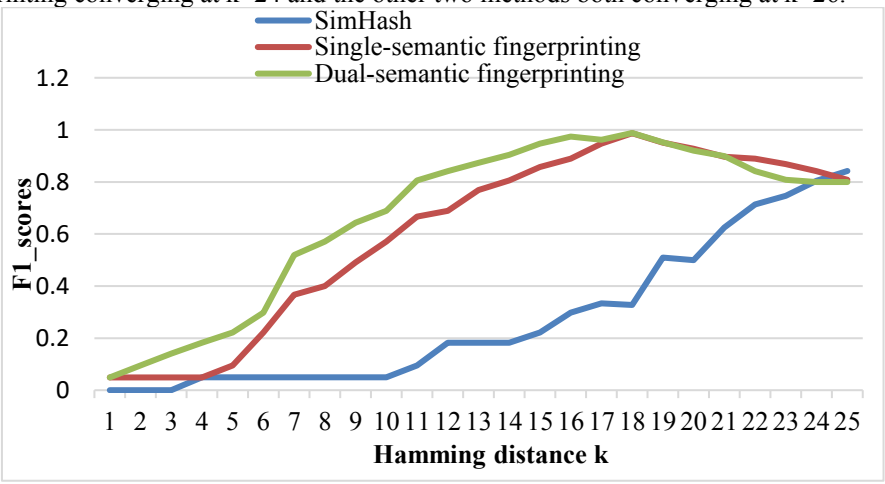

Fig. 2. F1_scores of the three methods at different thresholds

Obviously, the F1_score of the classical SimHash algorithm is significantly smaller than that of the other two algorithms. In addition, the F1_scores of the three methods converge when k>25, and the subsequent tests need only compare those in the range of k≤25.

The results of the Shapiro-Wilk test for normality in Table 2 show that F1_scores of two methods are not in normal distribution.

Table 2. Results of the Shapiro-Wilk normality test for the two methods

Detected methods	W statistic	P value
Single-semantic fingerprinting	0.8394	0.001116
Dual-semantic fingerprinting	0.8280	0.000693

Wilcoxon signed rank test was used to test the equivalence of the F1_scores. The P value of the Wilcoxon signed rank test for both single and dual semantic fingerprinting is 0.001068, which rejects the hypothesis that two F1_scores are equivalent.

Since the mean of F1_score of the dual-semantic fingerprinting method is greater than that of the single-semantic fingerprinting method, we conclude that the dual-semantic fingerprinting method outperforms the single-semantic method.

5 Conclusions

In this paper, we proposed a text similarity calculation method that introduced the CiLin path depth algorithm with the improved SimHash algorithm, which is different from the traditional text similarity calculation method and the classical SimHash algorithm. The proposed method is capable of detecting malicious text manipulation and realising infringement detection on large Chinese texts.

Although the experimental results show that the overall performance of the proposed method is satisfactory, there are still some limitations:

(1) The proposed method is more suitable for relatively long texts and does not give ideal results for short texts.

(2) Due to the size of the sample collection and the classification structure of CiLin, there are also some cases where the similarity calculation of some words is not accurate enough, and the time-consuming replacement of synonyms is unsatisfactory.

(3) Other features, such as the position of keywords in sentences, are not taken into account when calculating keyword weight.

(4) Without considering time complexity and efficiency, the effect of the keyword extraction ratio on the experimental results is also an issue worthy of further investigation.

In future studies, we will further explore the improvement of the digital fingerprinting technology in the semantic aspects of short texts. Meanwhile, the influence of more lexical structure attributes on the weight calculation will be considered, and the word replacement process will be improved to reduce the time consumption.

Acknowledgements

This research was funded by the Open Project of Key Laboratory of Knowledge Mining and Knowledge Services in Agricultural Converging Publishing, National Press and Publication Administration, grant number 2021kmks03.

REFERENCES

1. https://www.ncac.gov.cn/chinacopyright/upload/files/2020/9/16184424444.pdf
2. Wang C.L., Yang Y.H., Deng F., et al. (2019). A review of text similarity approaches. *Information Science*, 37, 158-168. https:// doi: 10.13833/j.issn.1007-7634.2019.03.026.

3. Peng J., Yang D.Q., Tang S.W., et al. (2008). A new similarity computing method based on concept similarity in Chinese text processing. *Science in China Series F: Information Sciences*, 51, 1215-1230. https://doi.org/10.1007/s11432-008-0103-4
4. Li Z., Lin C., Li B.C. (2009). Detection of repetitive reviews base on Hash technology. *Journal of Computer Applications,* 29, 263-266. https://doi:CNKI:SUN:JSJY.0.2009-S2-090
5. Charikar M. S. (2002). Similarity estimation techniques from rounding algorithms. In Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing. Montreal, Quebec, Canada. https://doi.org/10.1145/509907.509965
6. Manku G. S., Jain A., Das S. A. (2007). Detecting near-duplicates for web crawling. In Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada. https://doi.org/10.1145/1242572.1242592
7. Zhang H., Sheng Z.W., Zhang S.B., et al. (2020). Application of Simhash algorithm in text deduplication. *Computer Engineering and Applications*, 56, 246-251. https: //doi: 10. 3778/ j.issn.1002-8331.1902-0246
8. Gyawali B., Anastasiou L., Knoth P. (2020). Deduplication of Scholarly Documents using Locality Sensitive Hashing and Word Embeddings. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France. https: //doi. org/ 10. 1145/ 3459930.3469521
9. Yu Y., Hu Z., Zhang Y. (2015). Research on Large Scale Documents Deduplication Technique based on Simhash Algorithm. In Proceedings of the International Conference on Information Sciences, Machinery, Materials and Energy, Chongqing, China. https: //doi: 10. 2991/icismme-15.2015.262
10. Han X.G., Qu W., Yao X.X., et al. (2014). Research on malicious code variants detection based on texture fingerprint. *Journal on Communications*, 35, 125-136. https: //doi: 10. 3969 /j. issn.1000-436x.2014.08.016
11. Uddin M.S., Roy C.K., Schneider K.A., et al. (2011). On the effectiveness of Simhash for detecting near-miss clones in large scale software systems. In Proceedings of the 2011 18th Working Conference on Reverse Engineering, Lero, Limerick, Ireland. https: //doi. org/ 10. 1109 /wcre.2011.12
12. Dong B., Zheng Q.H., Song K.L., et al. (2011). Efficient Near-duplicate Detection Based on Multiple SimHash Fingerprints. *Journal of Chinese Computer Systems*, 32, 2152-2157. https://doi:CNKI:SUN:XXWX.0.2011-11-004
13. Liang Y., Tao Y., Feng, N., et al. (2017). Aggregating sentence-level features for Chinese near-duplicate document detection. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy. https: //doi. org/ 10. 1109/icnsc.2017.8000087
14. Hu H., Zhang L., Wu J. (2015). Hamming distance based approximate similarity text search algorithm. In Proceedings of the 2015 Seventh International Conference on Advanced Computational Intelligence (ICACI), Wuyi, China. https://doi.org/10.1109/icaci.2015.7184772
15. Sood S., Loguinov D. (2011). Probabilistic near-duplicate detection using Simhash. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, United Kingdom. https://doi.org/10.1145/2063576.2063737
16. Jiang Q., Sun M. (2011). Semi-supervised simhash for efficient document similarity search. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA. https://doi.org/10.1016/j.neucom.2012.06.035

17. Williams K., Wu J., Giles C.L. Simseerx: a similar document search engine. In Proceedings of the 2014 ACM Symposium on Document Engineering, Fort Collins, Colorado, USA, 16-19 September 2014. https://doi.org/10.1145/2644866.2644895
18. Luo W.J., Sun Z.H. (2014). The Secure Synonym Search over Encrypted Data Using Simhash. *Journal of Wuhan University* (Natural Science Edition), 60, 459-465. https: //doi: 10.14188/j.1671-8836.2014.05.014
19. Fu Z., Shu J., Wang J., et al. (2015). Privacy-preserving smart similarity search based on Simhash over encrypted data in cloud computing. *Journal of Internet Technology*, 16, 453-460. https://doi:10.6138/JIT.2015.16.3.20140918
20. Bai R.J., Wang X.D., Wang X.Y. (2013). Literature Similarity Detection Based on Digital Fingerprint. *Library and Information Service*, 57, 88-95. https://doi:10.7536/j.issn.0252-3116.2013.15.014.
21. Chen,G, Chen G., Wu D., et al. (2021). An improved Simhash algorithm based malicious mirror website detection method. *Journal of Physics: Conference Series,* 1971, 012067. ttps://doi.org/10.1088/1742-6596/1971/1/012067
22. Jin X., Zhang S., Liu J., et al. (2017). Research on Similarity Detection of Massive Text based on Semantic Fingerprint. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science, Guangzhou, China. https: //doi. org/ 10. 22323/ 1.300.0009
23. Buyrukbilen S., Bakiras S. (2012). Secure similar document detection with simhash. In Proceedings of the 9th VLDB Workshop on Secure Data Management, Istanbul, Turkey. https://doi.org/10.1007/978-3-319-06811-4_12
24. Pang S., Yao J., Liu T., et al. (2020). A text similarity measurement based on semantic fingerprint of characteristic phrases. *Chinese Journal of Electronics*, 29, 233-241. https: //doi.org/10.1049/cje.2019.12.011
25. Chen H.C., Li F., Zhu X.H., et al. (2016). A Path and Depth—Based Approach to Word Semantic Similarity Calculation in CiLin. *Journal of Chinese Information Processing*, 30, 80-88. https://doi:CNKI:SUN:MESS.0.2016-05-011
26. Miller G.A., Charles W.G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6, 1-28. https://doi.org/10.1080/01690969108406936
27. Cem B., Bilal A. (2022). Deep-Cov19-Hate: A Textual-Based Novel Approach for Automatic Detection of Hate Speech in Online Social Networks throughout COVID-19 with Shallow and Deep Learning Models. *Tehnički vjesnik*, 29(1), 149-156. https: //doi. org/ 10. 17559/TV-20210708143535
28. Salton G., Yang C.S., Yu C.T. (1975). A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science*, 26, 33-44. https: //doi. org/10.1002/asi.4630260106.
29. Mihalcea R., Tarau P. (2004). TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain. https://digital.library.unt.edu/ark:/67531/metadc30962/m1
30. Mei J.J., et al. CiLin (1st edition). (1983). Shanghai Lexicographical Publishing House: Shanghai, China.