Received: March 25, 2019

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Technique Using Multiple K-Shingling Based Weighted Dissimilarity Score for Web Content Outlier Mining

Liyakath AliKhan Raheemaa Khan[1]* **Mohammed Saleem Irfan Ahmed[2]**
Husni Hamad Almistarihi[2]

[1]*Bharathiar University, Coimbatore, India*
[2]*Department of Computer and Information Sciences,*
College of Science and Arts, Al Ula, Madinah, Saudi Arabia
* Corresponding author's Email: rlrkhan@gmail.com

Abstract: The technological evolution of Internet and Web along with several applications leads to the problem of redundancy as the documents are unanimously forwarded and are stored in several servers and platforms. Recently, not only duplicate documents but also near-duplicate documents affect the performance of the search results. The main objective of this paper is to provide significant documents by eliminating the redundancy and near redundancy documents present in the web search results. The proposed model comprises of two phases such as pre-processing phase and dissimilarity computation phase. For dissimilarity computation, the proposed model employs multiple k-shingling based dissimilarity score to identify the duplicate and near-duplicate documents which are considered as the outliers present in the set of input web documents. The proposed model has been evaluated using several experimental analysis. As there are no real datasets available for duplicate detection, datasets have been created and the performance evaluation is carried out with the created datasets. Several statistical analysis has been made wherein the average specificity, sensitivity, precision, and accuracy are 87%, 93%, 80%, and 92% respectively. The comparative analysis has also been made with various existing methods, in which the proposed model provides better results than existing methods in removing near-duplicates. The proposed multiple k-shingling based weighted dissimilarity model effectively detects the duplicates and near-duplicates when the number of outliers is minimum.

Keywords: Duplicates, Near-duplicates, Multiple K-shingling, Weighted dissimilarity score, Web content outlier mining.

1. Introduction

In recent years, the Web and the Internet become an indispensable tool for the human fraternity. Due to the increase in the usage of Internet and Web, the size of the web is getting increased with digitalization. Though there are plenty of advantages available in the Internet and Web, the increase in the digital documents leads to replication of information. This replicated or redundant documents cause a huge problem for search engines and web applications in extracting the information for web users. Not only duplicates but also near-duplicates causes a vast issue as the near-duplicate documents are much more similar to the original documents and differ only with a minimum text. The research on this near-duplicate detection captures more attention in recent days [1].

Apart from the size and noises present, the web also possesses several other complex characteristics such as dynamic and heterogeneity which plays an important role in mining the web data. Due to the characteristics of the web and the presence of noises, the web content extraction becomes a more complicated process. The noises present in the web contents are termed as web content outliers [2]. Eliminating the noises present in the web search results such as duplicates and near-duplicates become significant for the end user as it wastes the user's time by making them surf duplicate

documents present in several sites and also distracts the users surfing behaviour. Thus, by eliminating the noises, significant documents can be extracted and can be presented to the user.

This paper presents the novel approach for detecting duplicate and near-duplicate documents from the set of input web documents. The method uses multiple k-shingles represented as patterns from the input documents in which instead of using term frequency directly, the log frequency weighting and the length normalization are applied to the patterns as in cosine similarity measure for computing the dissimilarity between the documents. The dissimilarity scores that are minimum are considered as similar documents and are eliminated after comparing the documents with other documents.

The organization of the paper is as follows. Section 2 presents the literature survey related to noise removal from the web. Section 3 introduces the novel method with pre-processing phase and duplicate detection phase using multiple k-shingles based dissimilarity score in measuring the similarity between the documents along with the algorithm. The experimental analysis is presented in section 4. The results based on the experimental analysis are given in section 5. Finally, the paper concludes the proposed work in section 6.

2. Literature review

Several techniques have been developed in detecting duplicate and near-duplicates [3, 4]. Near-duplicate document identification from the particular domain was presented by Hajishirzi et al. [5]. In this method, the documents are represented as k-gram vectors and weights are optimized using improved cosine similarity or the Jaccard coefficient similarity measures. Also, the vectors are mapped to the small hash values that act as a document signature using locality sensitive hashing scheme. A partial duplicate detection was proposed that identify the partial duplication that exist in the same document using two subtasks such as sentence level near-duplicate detection and sequence matching [6]. However, the main drawback of this method is that it is applicable for small contents such as news articles and e-mail messages.

The n-gram based approach becomes the most popular and significant milestone in text mining. Also, to compute the relevancy score for the documents [7, 8], the methods assumed the existence of domain dictionary for mining web content outliers and the method also employs the vector space model [9]. Though the method provides good results, the main weakness is its high computational time. This issue was solved by introducing simple computation for detecting duplicates and extracting relevant documents using several mathematical concepts such as simple signed approach [10], set theoretical approach [11], linear correlation [12], statistical approach [13] and weighted approach [14] were introduced in effective web content outlier mining. The main downside is that the methods focus on single terms without considering term patterns due to which the accuracy get minimized.

Several techniques exist in extracting core contents from the web pages [15]. Tree edit distance was introduced in computing the text similarity between the syntactic n-grams and vector space model [16]. Several distance based similarity measures such as Dice's similarity coefficient, Cosine similarity, and Jaccard coefficient was compared using document fingerprint and it is proved that the cosine similarity provides the better result for Indonesian text [17]. These methods lack in fast processing.

Several mathematical concepts such as correlation metrics [18, 19], enhanced weighted approach [20], proximity based term frequency approach [21] was proposed in detecting outliers present in the web documents. For detecting near-duplicates from the set of web pages, Kumar et al., introduced sentence level features along with fingerprinting method that acts as cascade filters [22]. However, if the input is huge, the proposed method employs k-mode clustering before generating the fingerprint. A new search engine was developed termed as SimSeerX to extracts similar documents from the web using several similarity functions [23]. This model is useful in many applications such as plagiarism detection and near-duplicate detection. All these methods consider the irrelevant and duplicate documents as outliers and provide a common strategy that reduces the efficiency of the underlying model.

Several hashing techniques such as minhash [24], simhash [25] and hybrid hash [26] techniques are widely used to eliminate the noises present in the web pages and also extracting the duplicates and near-duplicate blocks present in the web pages. Noisy Data Cleaner (NDC) algorithm [27] was introduced to extract core content and to eliminate the noises present in the web pages. However, the method fails in detecting near duplicates. Thus, the research paper focuses on detecting duplicate and near-duplicates and on extracting core content.

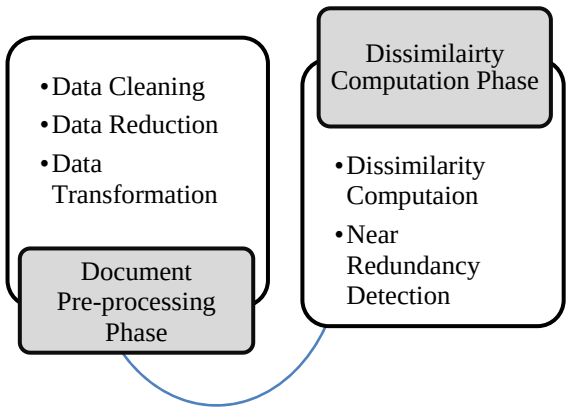

Figure. 1 The overall model of the proposed method of web content outlier mining

3. Proposed methodology

This section presents a novel technique for removing the duplicate documents present in the set of web documents. The method has two phases in which the first phase pre-processes the documents and the second phase computes the dissimilarity between the documents for detecting the duplicates and near-duplicates. The overall idea of the proposed web content outlier mining system is given in Fig. 1.

From Fig. 1, the proposed model is clustered into two processes in which the first one is the pre-processing step which is common for any document or information retrieval applications and cannot be avoided. The second phase is the redundancy detection phase which is intended to identify the duplicates and near duplicates that are considered as noises among the set of documents.

3.1 Pre-processing phase

Pre-processing phase is the first and leading process in any mining techniques as it improves the quality of the result produced by the mining process. The pre-processing phase includes various steps such as data cleaning, data reduction, and data transformation. The step prepares the data for mining interesting knowledge from the web documents for the user. The steps in the pre-processing phase are depicted in Fig. 2.

3.1.1. Data cleaning

As in information retrieval and text mining, the data cleaning step tries to remove the less significant features from the underlying dataset. The extracted document may include images and other types of data that are less important for mining. The various other types of data except text are removed initially.

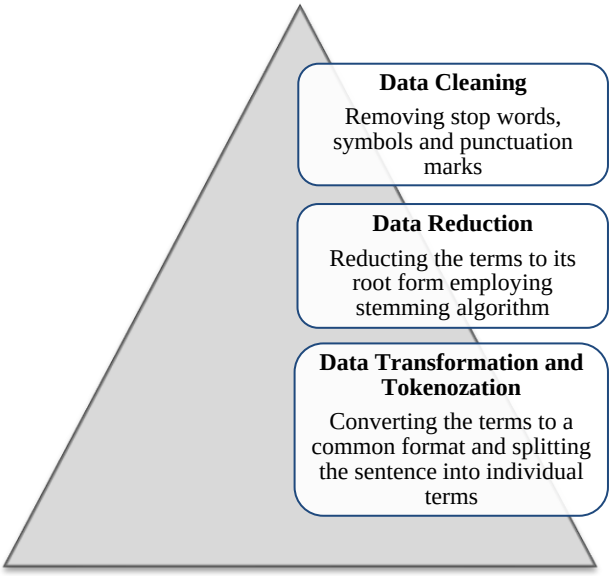

Figure. 2 Document pre-processing steps

In this busy world, information in the web is growing tremendously as the volume of data is updated daily on web. And most of the people use the internet search engine to find and retrieve the information.

a) Sample Input Text

busy information web growing tremendously volume data updated daily web people internet search engine retrieve information

b) Sample Output Text

Figure. 3 Data cleaning illustration

Also, the set of terms that represents the content of the document are processed in which the terms having less significant meaning are also eliminated. Stop words are generally referred to the most frequently used words such as *a*, *an*, *the* but conveys less meaning. There are approximately 500 stop words present in the English language which have low information value.

Thus removing the stop words minimizes the storage requirement and also speed up the underlying process. Basically, the search engine removes the stop words present in the query given by the user before processing it to improve the efficiency of the search process. The sample input text is taken from Khan et al., [28] for the data cleaning process and the output text after removing the stop words and other punctuation marks such as comma, full stop etc., is given in Fig. 3.

The proposed model employs the Porter stemmer algorithm which removes the suffixes present in the terms in which the output produced will not be a complete word [29]. The output

busy information web growing tremendously volume data updated daily web people internet search engine retrieve information

a) Sample Input Text

busi inform web grow tremend volum data updat daili web peopl internet search engin retriev inform

b) Sample Output Text

Figure. 4 Porter stemming algorithm illustration

produced by the data cleaning process is given as an input for the stemming process and the result is shown in Fig. 4. The algorithm works by stripping the suffixes at several steps based on the syllable length measure. The output is then given to the next pre-processing step.

3.1.2 Data transformation and tokenization

In this pre-processing step, the terms that are stemmed are converted to the common format. All the terms in the documents are converted to the small case letter. Also, the tokenization is performed on the input documents, which is the process of breaking down the stream of text components into elements or words called tokens. Usually, the white space or line breaks act as a separator and the words are fragmented individually. This list of tokens is served as an input for the proposed model.

3.2 Dissimilarity computation

Once the steps of the pre-processing phase are completed, the tokens are given as an input for the main phase of the proposed model to identify the duplicates and near-duplicates from the given set of input web documents. The workflow of the dissimilarity computation phase of the proposed model is depicted in Fig. 5. The pre-processed input web documents are given as an input for this phase.

The proposed method employs multiple k-shingles for computing the dissimilarity between the documents. Shingling is the most commonly used method that represents the document as a set. Based on the k value, the shingles group the set of words to a single component. Thus k-shingles represent a set of consecutive k terms in the documents. A K-shingle acts similar to the bag of words concept when the value of k is 1. Table 1 represents the shingles with k=1, 2, 3 that are computed for the sample input statement. The proposed method uses multiple k-shingles in which k varies from 1, 2, 3 and 4. Each input documents are processed and the shingles generated are represented as patterns. These

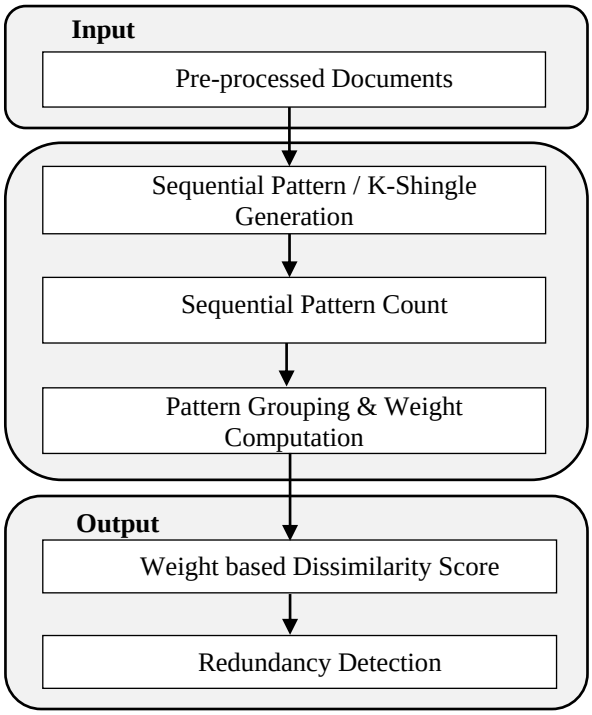

Figure. 5 The workflow of the dissimilarity computation phase of the proposed model

Table 1. Sample K-shingle generation

Input Statement	**New Delhi is the capital of India**
K = 1 (1-Shingle)	{[New], [Delhi], [is], [the], [capital], [of], [India]}
K = 2 (2-Shingle)	{[New Delhi], [Delhi is], [is the], [the capital], [capital of], [of India]}
K = 3 (3-Shingle)	{[New Delhi is], [Delhi is the], [is the capital], [the capital of], [capital of India]}

generated shingles are grouped based on the *k* values. The pattern counts for the shingles are computed for the input documents.

The weights are assigned for the pattern group by computing the ratio between the order of the pattern group to the maximum *k* value. For small documents, the maximum value of k can be 2, for news articles the maximum value of k can be taken as 3 and for large documents the maximum value of k can be taken as 4 [30]. In the proposed method, the maximum k value is 4 as it provides an effective result. Thus the weight for the 1-shingle pattern group (k = 1) is 1/4; the weight for the 2-shingle pattern group is 1/2; the weight for the 3-shingle pattern group is 1/3 and the weight for the 4-shingle pattern group is 4/4.

Once the weights are assigned for the pattern groups, the next step is to compute the dissimilarity score for the documents using the proposed weighted formula. The general formula to compute the weighted dissimilarity score between the two documents d and d' are given as in Eq. (1).

Table 2. Sample patterns and their frequency for the sample input documents

K-Shingles	Pattern ID	Patterns	Term Frequency		
			D1	D2	D3
1	P1	Web	5	6	8
1	P2	Content	9	7	7
1	P3	Outlier	8	5	9
1	P4	Mining	6	5	6
2	P5	web content	4	4	6
2	P6	content outlier	6	4	5
2	P7	outlier mining	5	3	5
3	P8	web content outlier	2	2	3
3	P9	content outlier mining	0	2	1

$$wt\_dissim(d,d') = \sum_{k=1}^{m} \frac{k}{m} \left(\sum_{i=1}^{n} |p_i - p_i'| \right) \quad (1)$$

where m is the number of shingles used in the model and for implementation the number of shingles (m) is taken as 4. d_i and d_i' are the normalized length weights of the pattern i in document d and d' respectively. The computation of normalized length weights is explained further.

Instead of using the term frequency directly, the log frequency of the patterns in the document are computed and are normalized based on the document length for efficient computation of weighted dissimilarity score between the documents. The log weight frequency of the pattern p in document d is computed as in Eq. (2).

$$wt(p,d) = \begin{cases} 1 + log_{10}(tf_{p,d}) & if\ tf_{p,d} > 0 \\ 0 & otherwise \end{cases} \quad (2)$$

where $tf_{t,d}$ is the term frequency of the pattern p in document d and the value of $weight(p,d)$ is $1 + log_{10}(tf_{p,d})$, if the frequency of the pattern is greater than 0 and 0 otherwise. Once the log weights for the patterns in the documents are computed, the length normalization of the pattern p in document d can be applied as given in Eq. (3).

$$Norm\_wt(p,d) = \frac{weight(p,d)}{\sqrt{\sum_{i=1}^{n} weight(p_i,d)^2}} \quad (3)$$

where t_i is the terms in the document d and $\sqrt{\sum_{i=1}^{n} Norm\_weight(t_i,d)^2} = 1$.

Finally, the weighted dissimilarity score is computed using the formula given in Eq. (1). The lower dissimilarity score that is nearer to the value 0 is said to be the duplicate or similar documents and the higher dissimilarity score implies that the documents are dissimilar. The threshold can be fixed in which the values less than 0.2 are considered as near-duplicates and can be removed.

The illustration for computing the dissimilarity score between the documents using multiple k-shingle based weighted dissimilarity score is explained with an illustration. In this example, the maximum k value is taken as 3. The illustration has been carried out with 3 documents. Table 2 shows the patterns with 1-shingles, 2-shingles, and 3-shingles along with the pattern ID. The frequency of patterns appeared in the three documents D1, D2, D3 are presented in Table 2.

The log weight frequency for the patterns in the documents D1, D2, and D3 along with the length normalization is computed for the values given in Table 2. The log weight frequency and length normalization calculation are computed as given in Eq. (3) and Eq. (4). The values are listed in Table 3. Finally, the patterns are grouped based on the k value and the final score for the documents are computed using weighted dissimilarity measure as given in Eq. (3). The values are presented in Table 4. From Table 4, the dissimilarity score for the documents D1 and D2 is 0.333. The score for documents D2 and D3 is 0.082 and for the documents D1 and D3 is 0.250. The dissimilarity score for the documents D2 and D3 is 0.082 which

Table 3. Log weight frequency and length normalization

Pattern ID	Patterns	Log Weight Frequency			Length Normalization		
		D1	D2	D3	D1	D2	D3
P1	Web	1.70	1.78	1.90	0.348	0.370	0.372
P2	Content	1.95	1.85	1.85	0.401	0.384	0.361
P3	Outlier	1.90	1.70	1.95	0.390	0.354	0.382
P4	Mining	1.78	1.70	1.78	0.365	0.354	0.348
P5	web content	1.60	1.60	1.78	0.328	0.334	0.348
P6	content outlier	1.78	1.60	1.70	0.365	0.334	0.332
P7	outlier mining	1.70	1.48	1.70	0.348	0.308	0.332
P8	web content outlier	1.30	1.30	1.48	0.267	0.271	0.289
P9	content outlier mining	0.00	1.30	1.00	0.000	0.271	0.196

Table 4. Multiple K-shingles based dissimilarity score

K-Shingles	W_Similarity (D1, D2)	W_Similarity (D2, D3)	W_Similarity(D1, D3)
1	0.014	0.000	0.013
2	0.044	0.025	0.019
3	0.275	0.057	0.218
Final Score	***0.333***	***0.082***	***0.250***

is very minimum than the threshold value 0.2. Thus to choose the near-duplicate document, the similarity score for the D2 and D3 is compared with other document D1.

The documents D3 and D1 is having less score as 0.25 and thus the document D3 is removed as it is considered as the near-duplicate documents. The algorithm for the proposed multiple K-Shingle based weighted dissimilarity score is given in Fig. 6.

4. Experimental setup

This section presents the dataset creation for the proposed system and various evaluation metrics used for the performance analysis and comparison with existing techniques.

4.1 Dataset creation

As there is no real time data set available for the web content outlier mining, the dataset has been created for the proposed model. This dataset includes revenant documents along with duplicates and near-duplicate documents extracted from the web. For dataset creation, 100 relevant documents termed as RD and 100 duplicate and near-duplicate documents termed as DD are extracted from the web. These documents form a base and based on which three different datasets have been created by varying the proportions of RD and DD.

Dataset I (DS1): The proportion of RD and DD is varied with a large number of RD than DD in a ratio of 75:25.

Dataset II (DS2): The proportion of RD and DD is varied with an equal number of RD and DD in a ratio of 50:50.

Dataset III (DS3): The proportion of RD and DD is varied with less number of RD than DD in a ratio of 25:75.

With these document datasets, the experiments have been performed with several trials by varying the number of relevant documents and duplicate documents.

Algorithm: Multiple K-Shingle based Weighted Dissimilarity Score
Input: Set of pre-processed documents D
Output: Near duplicate documents
FUNCTION wt_dissim_score(documents D)
threshold_value = 0.2
//Shingle generation & frequency count computation
For each document d in the input set
For k from 1 to m
For i from 1 to n
Compute k-shingles and their frequencies
pattern[i, d] = shingles
pattern_count[i, d] = frequency
End For
End For
End For
//Log Weight Frequency Computation
For each document d in the input set
For i from 1 to n
If pattern_count[i, d] > 0
weight[i, d] = 1 + log(pattern_count[i, d])
Else weight[i, d] = 0
End If
End For
End For
//Length Normalization Computation
For each document d in the input set
For i from 1 to n

$$\text{Norm\_wt}(p, d) = \frac{\text{weight}(p, d)}{\sqrt{\sum_{i=1}^{n} \text{weight}(p_i, d)^2}}$$

End For
End For
//Compute the Dissimilarity Score
For each document d in the input set
For each other document d' in the document set
For k from 1 to m
For each pattern p in the document

$$\text{wt\_dissim}(d, d') = \sum_{k=1}^{m} \frac{k}{m} \left(\sum_{i=1}^{n} |d_i - d_i'| \right)$$

End For
End For
End For
End For
//Duplicate document extraction
For all the documents d & d' in the document set.
If wt_dissim(d,d') < threshold_value
Fetch the duplicate documents
End If
Compare the minimum dissimilarity score d with all other documents and the minimum dissimilarity score d' with all other documents
Mark and fetch the document d or d' having minimum dissimilarity score
End For
END FUNCTION

Figure. 6 Multiple K-shingle based weighted dissimilarity score

4.2 Evaluation metrics

Several evaluation metrics are used in evaluating the performance of the proposed model. The measures used in evaluating the performance of the proposed system are explained below.

True Positive: It is the count of the number of duplicates documents that are correctly predicted as duplicates by the proposed model.
False Positive: It is the count of the number of duplicates documents that are incorrectly predicted as relevant by the proposed model.
True Negative: It is the count of the number of relevant documents that are correctly predicted as relevant documents by the proposed model.
False Negative: It is the count of the number of relevant documents that are incorrectly predicted as duplicate documents by the proposed model.
Sensitivity: It measures the ratio of correctly identified duplicate documents to the number of duplicate documents present in the underlying dataset. The formula to compute the sensitivity is given in Eq. (4).

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

Specificity: It measures the ratio of correctly identified relevant documents to the number of relevant documents present in the underlying dataset. The formula is given in Eq. (5).

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

Precision: It measures the ratio of correctly predicted duplicate documents to the number of predicted duplicate documents as given in Eq. (6).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Accuracy: It computes the overall prediction rate by calculating the ratio of correct results to the total number of documents as given in Eq. (7).

$$Accuracy = \frac{TP + TN}{TTP + TN + FP + FN} \tag{7}$$

F1-Measure: It is measured by computing the weighted harmonic mean of the precision and recall values. The formula is given in Eq. (8).

$$F1 - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

5. Performance analysis

The proposed method has been applied on the three datasets such as DS1, DS2, DS3 by varying the dataset sizes and the details are given in Table 5. In Table 5, DS1-1 to DS1-10 represents the trials using DS1 Dataset. DS2-1 to DS2-10 represents the trials using DS2 dataset and DS3-1 to DS3-10 represents the trials using DS3 dataset. Analysis has been made by the number of duplicate documents extracted by the proposed model and by which the sensitivity, specificity, precision, accuracy, and F1-measure are computed and are shown in Table 5.

Thus, the average percentage of sensitivity, specificity, precision, accuracy, and F1-score for the proposed model with the dataset DS1 are 94.20%, 95.35%, 87.01%, 95.06%, and 90.41% respectively, whereas, the average percentage of sensitivity, specificity, precision, accuracy, and F1-score for the proposed model with the dataset DS2 are 85.08%, 91.74%, 76.85%, 90.04%, and 80.49% respectively. Similarly, the average percentage of sensitivity, specificity, precision, accuracy, and F1-score for the proposed model with the dataset DS3 are 82.75%, 91.60%, 76.01%, 89.38%, and 78.95% respectively. From the result analysis made from the three datasets, the proposed method provides a higher classification accuracy and better result when there is a minimum number of duplicate documents to be classified.

The comparative analysis has also been made with the proposed method by comparing it with other existing methods such as N-gram approach [7], sentence level features with fingerprints (SLF-FP) [22], SimSeerX [23], enhanced weighted approach [19], Simhash [25], hybrid hash [26], and NDC algorithm [27]. The comparative analysis for the proposed model and the other mentioned exiting model is carried out and 100 documents from the three datasets DS1, DS2, and DS3 are taken for the analysis having varied number of RD and DD documents. The values for the sensitivity, specificity, precision, accuracy and F1-score for various methods are given in Table 6. From Table 6, sensitivity, precision and F1-score values of the N-gram are very low when compared with other methods.

However, the methods such as SLF-FP, SimSeerX, Simhash, Hybrid Hash, and NDC algorithm provides a better result in detecting relevant documents than detecting duplicate

Table 5. Experimental analysis for the proposed model

Trial ID	Dataset Size	No. of RD	No. of DD	True Positive	False Positive	True Negative	False Negative	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score (%)
DS1-1	10	8	2	2	0	8	0	100	100	100	100	100
DS1-2	20	15	5	5	1	14	0	100	93.33	83.33	95.00	90.91
DS1-3	30	23	7	7	1	22	0	100	95.65	87.50	96.67	93.33
DS1-4	40	30	10	9	2	28	1	90.00	93.33	81.82	92.50	85.71
DS1-5	50	38	12	11	2	36	1	91.67	94.74	84.62	94.00	88.00
DS1-6	60	45	15	14	2	43	1	93.33	95.56	87.50	95.00	90.32
DS1-7	70	53	17	16	3	50	1	94.12	94.34	84.21	94.29	88.89
DS1-8	80	60	20	18	3	57	2	90.00	95.00	85.71	93.75	87.80
DS1-9	90	68	22	20	3	65	2	90.91	95.59	86.96	94.44	88.89
DS1-10	100	75	25	23	3	72	2	92.00	96.00	88.46	95.00	90.20
DS2-1	10	8	2	2	1	7	0	100	87.50	66.67	90.00	80.00
DS2-2	20	15	5	4	2	13	1	80.00	86.67	66.67	85.00	72.73
DS2-3	30	23	7	6	2	21	1	85.71	91.30	75.00	90.00	80.00
DS2-4	40	30	10	8	3	27	2	80.00	90.00	72.73	87.50	76.19
DS2-5	50	38	12	10	3	35	2	83.33	92.11	76.92	90.00	80.00
DS2-6	60	45	15	12	3	42	3	80.00	93.33	80.00	90.00	80.00
DS2-7	70	53	17	14	3	50	3	82.35	94.34	82.35	91.43	82.35
DS2-8	80	60	20	17	4	56	3	85.00	93.33	80.95	91.25	82.93
DS2-9	90	68	22	19	4	64	3	86.36	94.12	82.61	92.22	84.44
DS2-10	100	75	25	22	4	71	3	88.00	94.67	84.62	93.00	86.27
DS3-1	10	8	2	2	1	7	0	100	87.50	66.67	90.00	80.00
DS3-2	20	15	5	4	2	13	1	80.00	86.67	66.67	85.00	72.73
DS3-3	30	23	7	5	2	21	2	71.43	91.30	71.43	86.67	71.43
DS3-4	40	30	10	8	3	27	2	80.00	90.00	72.73	87.50	76.19
DS3-5	50	38	12	10	3	35	2	83.33	92.11	76.92	90.00	80.00
DS3-6	60	45	15	12	3	42	3	80.00	93.33	80.00	90.00	80.00
DS3-7	70	53	17	14	3	50	3	82.35	94.34	82.35	91.43	82.35
DS3-8	80	60	20	16	4	56	4	80.00	93.33	80.00	90.00	80.00
DS3-9	90	68	22	19	4	64	3	86.36	94.12	82.61	92.22	84.44
DS3-10	100	75	25	21	5	70	4	84.00	93.33	80.77	91.00	82.35

Table 6. Comparative analysis for the proposed model with existing techniques

Various Techniques	True Positive	False Positive	True Negative	False Negative	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	F1-Score (%)
Dataset 1 (DS1) : Size – 100 Documents, Relevant -75 Documents, Duplicates – 25 Documents									
N-Gram	15	11	64	10	60.00	85.33	57.69	79.00	58.82
SLF-FP	18	7	68	7	72.00	90.67	72.00	86.00	72.00
SimSeerX	22	6	69	3	88.00	92.00	78.57	91.00	83.02
NDC	21	7	68	4	84.00	90.67	75.00	89.00	79.25
Weighted Approach	20	5	70	5	80.00	93.33	80.00	90.00	80.00
Simhash	20	7	68	5	80.00	90.67	74.07	88.00	76.92
Hybrid Hash	21	6	69	4	84.00	92.00	77.78	90.00	80.77
Proposed	23	3	72	2	92.00	96.00	88.46	95.00	90.20
Dataset 2 (DS2) : Size – 100 Documents, Relevant -50 Documents, Duplicates – 50 Documents									
N-Gram	14	13	62	11	56.00	82.67	51.85	76.00	53.85
SLF-FP	17	8	67	8	68.00	89.33	68.00	84.00	68.00
SimSeerX	20	6	69	5	80.00	92.00	76.92	89.00	78.43
NDC	19	7	68	6	76.00	90.67	73.08	87.00	74.51

Weighted Approach	20	6	69	5	80.00	92.00	76.92	89.00	78.43
Simhash	20	8	67	5	80.00	89.33	71.43	87.00	75.47
Hybrid Hash	21	6	69	4	84.00	92.00	77.78	90.00	80.77
Proposed	22	4	71	3	88.00	94.67	84.62	93.00	86.27
Dataset 3 (DS3) : Size – 100 Documents, Relevant -25 Documents, Duplicates – 75 Documents									
N-Gram	13	14	61	12	52.00	81.33	48.15	74.00	50.00
SLF-FP	17	9	66	8	68.00	88.00	65.38	83.00	66.67
SimSeerX	19	7	68	6	76.00	90.67	73.08	87.00	74.51
NDC	18	8	67	7	72.00	89.33	69.23	85.00	70.59
Weighted Approach	19	8	67	6	76.00	89.33	70.37	86.00	73.08
Simhash	18	10	65	7	72.00	86.67	64.29	83.00	67.92
Hybrid Hash	20	7	68	5	80.00	90.67	74.07	88.00	76.92
Proposed	21	5	70	4	84.00	93.33	80.77	91.00	82.35

documents as the specificity values are much higher than the corresponding sensitivity values. The weighted approach delivers a better result than all the existing method however, the proposed method gives even better values and the accuracy is above 90% for all the datasets.

Thus from the experimental analysis, it is clear that the proposed method provides better performance than the existing methods.

Also, the method gives much better results when the duplicate documents are low in par with the relevant documents. This is quite natural in the real world applications as the number of duplicate documents to be detected are very minimal when compared with the relevant documents. Thus, the method effectively identifies and removes the duplicate and near-duplicate documents from the set of input web documents.

6. Conclusion

This research work focuses on extracting the redundancy and near redundancy documents present in the set of web documents thereby providing the significant results to the user. The model has been proposed which computes the dissimilarity score between the documents using multiple k-shingles to identify the duplicate and near-duplicate documents that are considered as the outliers. The method uses log frequency weight and length normalization instead of using term frequency directly. To prove the efficiency of the proposed method, several experimental analysis and evaluation measures have been carried out. Three datasets have been created and various trials by varying the number of duplicate and relevant documents have been performed. Based on the results from the analysis, the average sensitivity and accuracy of the proposed method with a minimum number of duplicate documents are 94.20% and 95.06% and for the equal number of duplicates and relevant documents, the average sensitivity and accuracy of the proposed method are 85.08% and 90.04% whereas, for the maximum number of duplicate documents, the average sensitivity and accuracy of the proposed method are 82.75% and 89.38%. Thus, the proposed method detects the duplicates and near-duplicates from the set of input web document having a minimum number of outliers where the situation is obvious for many real-world applications.

References

[1] K. Muthmann, W. M. Barczynski, F. Brauer, and A. Loser, "Near-duplicate detection for web-forums" In: *Proc. of the International Database Engineering & Applications Symposium*, pp.142-151, 2009.

[2] M. Agyemang, K. Barker, and R. S. Alhajj, "Framework for mining web content outliers", In: *Proc. of the ACM symposium on Applied computing*, pp.590-594, 2004.

[3] Z. Tian, H. Lu, W. Ji, A. Zhou, and Z. Tian, "An n-gram-based approach for detecting approximately duplicate database records", *International Journal on Digital Libraries*, Vol.5, No.3, pp.325–331, 2001.

[4] G. S. Manku, A. Jain, and A. D. Sarma, "Detecting near-duplicates for web crawling" *In: Proc. of the International Conference on World Wide Web*, pp.141-150, 2007.

[5] H. Hajishirzi, W. T. Yih, and A. Kolcz, "Adaptive near-duplicate detection via similarity learning", In: *Proc. of the International Conference on Research and Development in Information Retrieval*, pp.419-426, 2010,

[6] Q. Zhang, Y. Zhang, H. Yu, and X. Huang, "Efficient partial-duplicate detection based on sequence matching", In: *Proc. of the International conference on Research and Development in Information Retrieval*, pp. 675-682, 2010.

[7] M. Agyemang, K. Barker, and R. S. Alhajj, "Mining web content outliers using structure oriented weighting techniques and N-grams", In: *Proc. of the ACM symposium on Applied computing*, pp. 482-487, 2005.

[8] M. Agyemang, K. Barker, and R. S. Alhajj, "Hybrid approach to web content outlier mining without query vector", *Data Warehousing and Knowledge Discovery*, pp.285–294, 2005.

[9] M. Agyemang, K. Barker, and R. S. Alhajj, "A comprehensive survey of numeric and symbolic outlier mining techniques", *Intelligent Data Analysis*, Vol.10, No.6, pp.521–538, 2006.

[10] G. Poonkuzhali, K. Thiagarajan, K. Sarukesi, and G. V. Uma, "Signed approach for mining web content outliers", *World Academy of Science, Engineering and Technology*, Vol.56, No.09, pp.820-824, 2009.

[11] G. Poonkuzhali, K. Thiagarajan, and K. Sarukesi, "Set theoretical Approach for mining web content through Outliers detection", *International Journal on Research and Industrial Applications*, Vol.2, pp.131-138, 2009.

[12] G. Poonkuzhali, R. K. Kumar, R. Krip Keshav, P. Sudhakar, and K. Sarukesi, "Correlation Based Method to Detect and Remove Redundant Web Document", *Advanced Materials Research*, Vol.171, pp.543-546, 2011.

[13] G. Poonkuzhali, R. K. Kumar, R. K. Keshav, K. Thiagarajan, and K. Sarukesi, "Effective Algorithms for Improving the Performance of Search Engine Results", *International Journal of Applied Mathematics and Informatics*, Vol.5, No.3, pp.216-223, 2011.

[14] G. Poonkuzhali, "Web Content Outlier Mining through Mathematical Approach*", Ph.D. Thesis*, Anna University, Chennai, 2011.

[15] S. Sirsat, "Extraction of Core Contents from Web Pages", *International Journal of Engineering Trends and Technology*, Vol.8, No.9, pp.484-489, 2014.

[16] G. Sidorov, H. Gómez-Adorno, I. Markov, D. Pinto, and N. Loya, "Computing text similarity using tree edit distance", In: *Proc. of Annual Conference of the Fuzzy Information Processing Society and World Conference on Soft Computing*, pp.1-4, 2015.

[17] T. Mardiana, T. B. Adji, and I. Hidayah, "The Comparation of Distance-Based Similarity Measure to Detection of Plagiarism in Indonesian Text", In: *Proc. of International Conference on Soft Computing, Intelligence Systems, and Information Technology*, pp.155-164, 2015.

[18] S. Sathya Bama, M. S. Irfan Ahmed, and A. Saravanan, "A Mathematical Approach for Improving the Performance of the Search Engine through Web Content Mining", *Journal of Theoretical & Applied Information Technology*, Vol.60, No.2, pp.343-350, 2014.

[19] S. S, Bama, M. S. I Ahmed, and A. Saravanan, "Enhancing the Search Engine Results through Web Content Ranking" *International Journal of Applied Engineering Research*, Vol.10, No.5, pp.13625-13635, 2015.

[20] S. Sathya Bama, M. S. Irfan Ahmed, and A. Saravanan, "A Mathematical Approach for Mining Web Content Outliers using Term Frequency Ranking", *Indian Journal of Science and Technology*, Vol.8, No.14, 2015.

[21] S. Sathya Bama, M. S. Irfan Ahmed, and A. Saravanan, "Relevance Re-ranking Through Proximity Based Term Frequency Model", In: *Proc. of International Conference on ICT Innovations*, pp.219-229. 2016.

[22] J. P. Kumar and P. Govindarajulu, "Near-duplicate web page detection: An efficient approach using clustering, sentence feature and fingerprinting", *International Journal of Computational Intelligence Systems*, Vol.6, No.1, pp.1-13, 2013.

[23] K. Williams, J. Wu, and C. L. Giles, "Simseerx: a similar document search engine" In: *Proc. of the ACM Symposium on Document Engineering*, pp.143-146, 2014

[24] A. Z. Broder, "Identifying and filtering near-duplicate documents", In: *Proc. of Symposium on Combinatorial Pattern Matching*, pp.1–10, 2000.

[25] P. Sivakumar, "Effectual web content mining using noise removal from web pages" *Wireless Personal Communications*, Vol.84, No.1, pp.99-121, 2015.

[26] R. Uma and B. Latha, "Noise elimination from web pages for efficacious information retrieval", *Cluster Computing*, pp.1-20, 2018.

[27] P. Sahoo and R. Parthasarthy, "An efficient web search engine for noisy free information retrieval", *International Arab Journal of Information Technology*, Vol.15, No.3, pp.412-418, 2018.

[28] M. R. L. Khan, M. I. Ahmed, and M. A. Riyad, "A novel analytical approach for identifying outliers from web documents", *International Journal of Applied Engineering Research*, Vol.12, No.22, pp.12156-12161, 2017.

[29] M. F. Porter, “An algorithm for suffix stripping”, *Program*, Vol.*14, No.*3, pp.130-137, 1980.
[30] J. Phillips, “Jaccard, Similarity and Shingling” *Data Mining*, University of Utah, 2013.