Research Article

Semantic similarity and text summarization based novelty detection

Sushil Kumar[1] · Komal Kumar Bhatia[1]

Received: 13 September 2019 / Accepted: 21 January 2020 / Published online: 4 February 2020

Abstract
Current web crawlers search the queries at very high speed, but the problem of novelty detection or redundant information still persists. It consumes precious time and memory of users in search of the new document over the internet. In this paper, an innovative novelty detection mechanism is proposed, which can be appended with the current web crawlers. The proposed mechanism first summarizes the text, based on ontology, and from the obtained summary, semantic similarity is calculated using word net 3.0. The hash value is then calculated using the winnowing algorithm. This hash value of the document is matched with others using the Dice coefficient to calculate the similarity index. Based on the threshold chosen for similarity, the document is treated either as novel or not. This proposed mechanism is implemented using SQL as backend and visual studio-2012 as frontend. The results show that the projected strategy not only reduces memory consumption but also decreases the number of resultant documents, hence minimize the user time for searching the data from the obtained results. Also, the proposed approach can be used with other search engines like Google, Yahoo, Bing, and Alta Vista to minimize superfluous documents.

Keywords Dice coefficient · Hash value · Novelty detection · Semantic similarity · Word net 3.0

1 Introduction

Novelty detection [1] is the process of finding information that has not appeared before, or new concerning the relevant information already seen. The information accumulated progressively due to the explosive growth of document over web, which has resulted in duplication of information. It consumes precious time and space of the user in reading the new information. Given below are three simple documents in a time sequence:

D_1: Singapore is an island city-state located at the southern tip of the Malay Peninsula. It lies 137 km north of the equator.
D_2: Singapore is an island city-state located at the southern tip of the Malay Peninsula. The population of Singapore is approximately 4.86 million.
D_3: Singapore lies 137 km north of the equator. The population is approximately 4.86 million.

When the general novelty detection method is directly applied, the document D_3 is quite natural to be predicted as a novel because it contains new information in comparison to D_1 and D_2 individually. But if these documents are segmented into sentences, D_3 will be correctly predicted as redundant because all its sentences have appeared in the previous sentences.

The problems associated with the current search engine and crawlers [2] are as listed below:

- One issue with a focused crawler [2] is that they miss essential pages by just creeping pages that are relied upon to give immediate benefits.
- The crawlers download numerous unimportant pages that lead to the utilization of system transfer speed.

✉ Sushil Kumar, panwar_sushil2k@yahoo.co.in; Komal Kumar Bhatia, komal_bhatia1@rediffmail.com | [1]Department of Computer Engineering, J.C. Bose University of Science and Technology, YMCA, Faridabad, Haryana 121006, India.

They receive pooling technique for the upkeep of freshness of database.
- The collaborative crawler [2] utilizes the gathering of crawling nodes; it is conceivable that several nodes download a similar page many times. Therefore needs to build up a method that decreases these cover of pages.
- The parallel crawler [2] having many crawling methods called C-Proc's. At the point when various C-Proc's are working freely, it is conceivable that more than one may download a similar page on many times.

To resolve the above said setbacks this paper introduces a novelty detection technique which overcomes the concern of downloading these redundant pages. The rest of the paper is organized as shown; Sect. 2 presents the work done in the area of novelty detection in text documents. Discussion of the developed software for the generic crawler is presented in Sect. 3. Proposed mechanism is presented in Sect. 4 followed by its implementation details in Sects. 5 and 6. Comparative analysis for generic crawler with proposed crawler is presented in Sect. 7, and finally, conclusion is given in Sect. 8 followed by references.

2 Related work

Many authors have carried out work on novelty detection research in the text document. The below mention Table 1 shows the contribution of various authors in the area of novelty detection [3, 4] and the comparison with the proposed methodology.

As seen in Table 1 most of existing methods only use the text similarities between the text documents. But this approach is not sufficient to remove redundancy problem in search results. So, there is a severe need to use such an approach, which can conquer this concern to large extent. This paper is an effort to propose a novelty detection mechanism to treat the above issue. The hallmarks of the proposed technique are:

- This approach uses text summarization of the document, and the obtained data is further summarized using the ontology of that domain. The semantic similarity is calculated using Word net 3.0 and similarity has calculated by the Winnowing fingerprint matching algorithm.
- This work focused on whether the incoming document contains new information with respect to the relevant information already stored in to reader memory or relevant reference. If the source sentence is 'I am Ram' and the target sentence is 'My name is Ram' then according to this approach both the sentences are treated as one due to conveying the same information. Such redundancy does not process by present state of art as text matching techniques.
- This approach extract semantic features from the target document concerning the source document by using word classes, synonyms, antonyms, and lexical databases such as word net. We report promise results with our features on the developed system. In this work similarity index is better than the Generic crawler approach.

3 Generic crawler methodology

In this work, firstly, a generic crawler is proposed that takes a query on a specific domain, and the crawler results are stored in an indexed database. The database stores the URL of the query together with its HTML tags, metadata tags, etc. The URL enter by the admin are stored in the dictionary. This method also provides a search interface on which the user can apply a query based on a specific domain, stored in the database. When the user types a keyword on the search interface, it shows the web pages stored in the database. The retrieved list of web pages may contain relevant and redundant results, which is a time-consuming task for the user to read all the pages. The architecture of generic crawler shown in Fig. 1.

Figure 2 shows the interface for domain-specific Generic Crawler, which includes website category related to education, politics, sports, technology, health, entertainment, travel, and zoology. Users can enter the website related topic and website link, as shown in the interface. Then the user clicks on the crawl button, and the generic crawler crawls the web pages based on the website URL (Uniform Resource Locator) to store them into the database.

Figure 3 shows the SQL database, which includes three tables T_Categoty, T_website, and T_webpages. The table T_Category includes the website categories i.e., education, politics, sports, technology, health, entertainment, travel, and zoology. The T_Website and T_Webpages store the website related information together with web pages related information. Upon executing the query by select command, the information from the database table is shown in the screenshot. The database stores the URL of the query together with its HTML tags, metadata tags, etc.

4 Proposed crawler methodology for novelty detection

In the proposed methodology, the limitation of a generic crawler that is repeated occurrence of the redundant documents is eliminated. The projected method provides the

Table 1 Contribution of work in the area of novelty detection

References	Methodology adopted
Allan et al. [5]	On temporal summaries of new topics, the task is to extract a single sentence from each event within a news topic, where the stories are presented one at a time and sentences from a story must be ranked before the next story can be considered
Allan et al. [6], Zhang et al. [4], Li and Craft [7], Kwee [8] and Bentivogli et al. [9], Zhao et al. [3], Zhang et al. [4]	Novelty and redundancy detection in adaptive filtering, the goal is to eliminate redundant documents Textual entailment based sentence level novelty mining was explored in the novelty subtask of RTE-TAC 6 and 7
Soboroff and Harman [10]	The concept of novelty detection from the text for novelty track in TREC. The TREC experiments were designed to retrieved relevant and novel sentences from a corpus based on a given query
Stokes and Carthy [11], Franz et al. [12], Brant et al. [1], Soboruff and Harmarn [10]	Numerous approaches with TDT to detect a new event for corpus. Current techniques for new event detection are usually based on clustering algorithms
Larkey et al. [13], Tsai et al. [14], Soboroff and Harman [10]	Research in novelty detection from text had been carried out at the event level, sentence level, and document level. Topic detection and tracking (TDT), where the main concern was to detect a new story from the series of new stories. Tsai et.al. represented sentences as vectors and computed similarity scores for a sentence currently being considered with each previous sentence with the cosine similarity function for detecting novel sentences
Li and Craft [7]	The motivation of the work on information-pattern-based novelty detection. Identifying query-related named entities
Alqaraleh and Ramadanin [15, 16]	The objective is to improve the efficiency of multimedia search engines by eliminating repeated occurrences. The software developed in this paper can work and create multimedia database files like images, music, and videos. But in our work, the novelty detection at textual document has been carried out, which involves the documents from different domains. The main difference in this approach is summarization [23] of the text document, and the obtained data is further summarized using the ontology [21, 22] of that domain. The semantic similarity [24] is calculated using word net 3.0 [25], and similarity has calculated by the Winnowing fingerprint matching algorithm [26]
Sravanthi and Srinivasu [17]	The word co-occurrence method ignores the word order of the sentence and does not account for the meaning of the word in the context of the sentence
Karkali [18], Dasgupta and Dey [19]	The problem of novelty detection at the document level has been carried out
Ghosal et al. [20]	The problem of novelty detection at document level has been carried by creating the resource via event-specific crawling of news documents across several domains in a periodic manner. The supervised machine learning approach is used in the system.TAP DND 1.0 corpus is used

relevant and novel results to the user and filters out the redundant ones. This work includes extractive text summarization using ontology to calculate the summary of the text document after that the Winnowing fingerprint algorithm [26, 27] is applied for similarity calculation and Word Net 3.0 [28] for semantic similarity [32, 33]. Winnowing calculation is a technique for word comparability search in a document by looking at the fingerprint on the document. The algorithm input is the text document, which is processed and yield as a hash value. The hash value is then called as the unique finger impression, which is utilized to look at the comparability of each document. The difference of Winnowing Calculation and another calculation of comparability indicator is in the choice procedure of its fingerprint. The consequence of hash value figuring is partitioned into window w in which the smallest value will be taken from every window for the document fingerprint. The stepwise procedure for proposed mechanism is given below:

- Firstly, the text summarization technique is applied using ontology, which provides the relevant sentences.
- Both the target text and the original text are assumed as string s with the length t.
- N-grams [29, 30] are generated from tokens to obtain documents with fixed-length strings.
- N-grams [29, 31] are further processed for discovering hashes, which are collected in order to diminish the size of documents.

Fig. 1 Generic web crawler

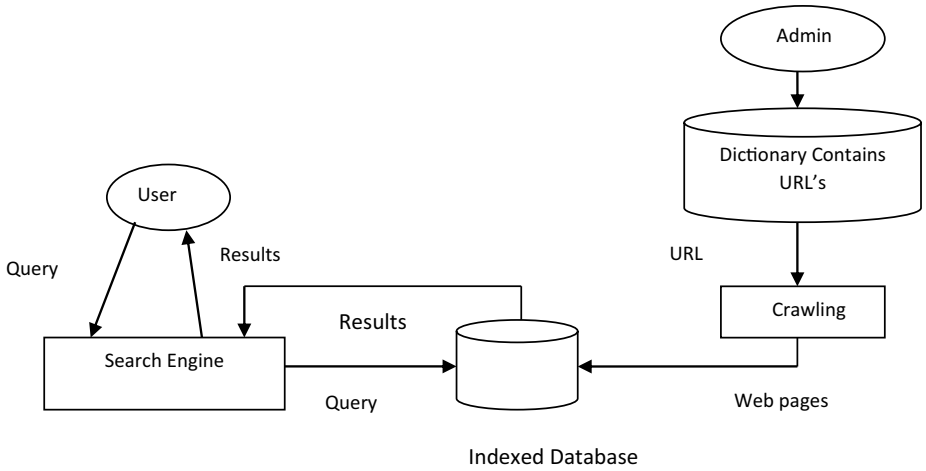


Fig. 2 Generic crawler novelty

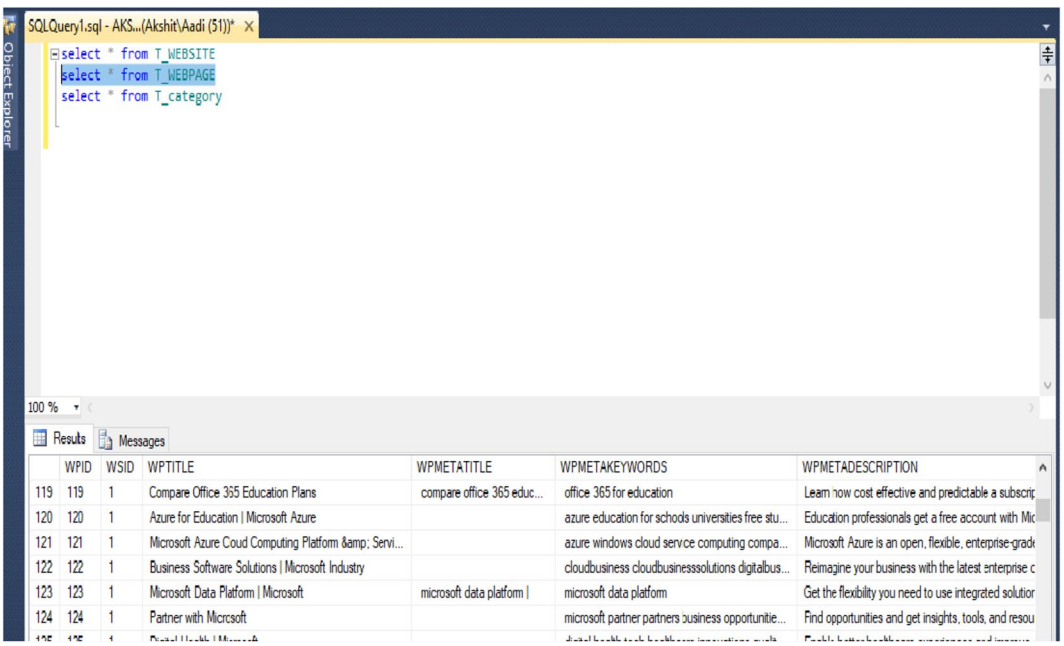


Fig. 3 SQL indexed database

- The strings are reformed to some numeric values called hashes [34]. A suitable similarity measure is applied to hashes for similarity determination.

4.1 Algorithm for the proposed crawler novelty detection

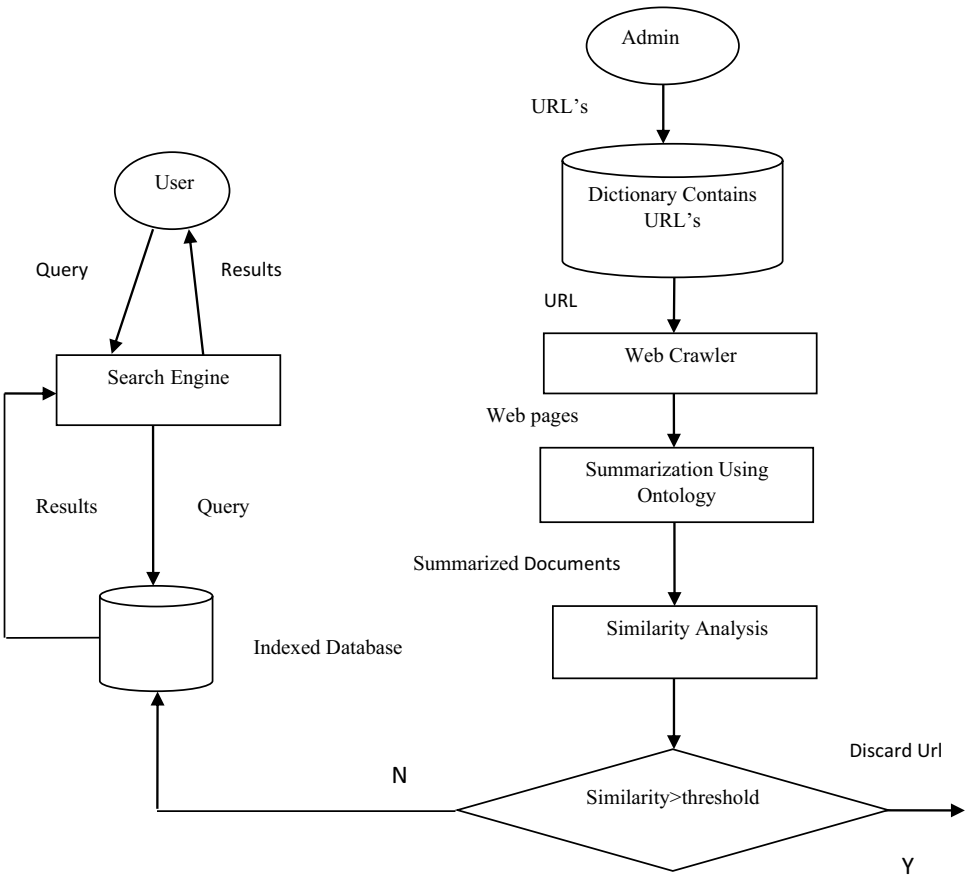
Input: URL (source code in S_out), DB-> Data Base
DB first row say in S_current
Begin
Step 1: Fetch source code in S_out
Step 2: Summarization of fetched data S-out
Step 3: for each row in DB
3.1 Summarization of each row of DB (S_current)
Step 4: Find the similarity of both summarized data (S_out , S_current)
Step 5: if (similarity > threshold)
5.1 Break the loop and will not compare with any row
5.2 Because it already finds a similar row
Else Check with another DB row
Step 6: If it does not find the similar doc in DB, than save a new row i.e. fetched data in DB.
End.

The architecture of the proposed methodology is shown in Fig. 4.

4.2 Detailed steps for proposed crawler novelty

In Fig. 5, Steps of the ontology-based text summarization are shown. It consists of sentence parsing, tokenization, and stopword removal, noun filtering using WorldNet 3.0,

Fig. 4 Proposed architecture



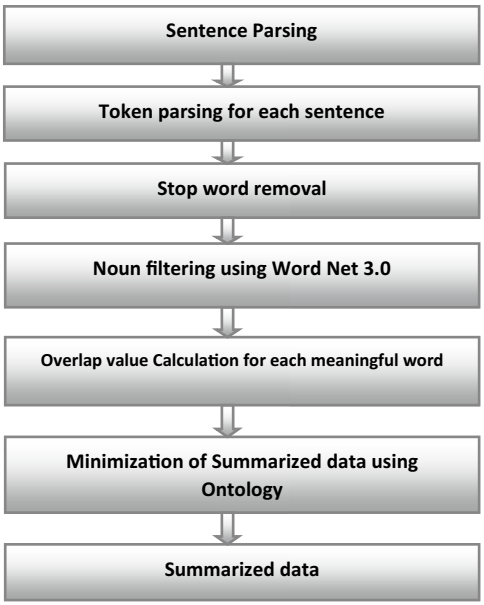


Fig. 5 Ontology based text summarization

word overlap calculation and minimization of summarized data using ontology [21, 22]. These steps are explained in brief as below:

4.2.1 Sentence parsing

A document consists of a large number of sentences, so the document is broken into sentences. The sentences consist of a large number of words, and it is not always necessary that each word is of importance. Due to which high dimensionality of the document has to be reduced by processing the document to get rid of extra words to obtain the weight of each of the word to be used in the algorithm.

4.2.2 Tokenization

The processes involved in information retrieval require the words of documents. Tokenization is used to identify meaningful words called tokens. The primary use of tokenization is to split the sentences into individual tokens. For example 'There are readers who prefer learning' so in this sentence 'there', are, 'readers', 'who', 'prefer' and learning' are the tokens.

4.2.3 Stop-words removal

Stop words are frequently occurring, insignificant words that appear in a database record, article, or a web page, etc. pronoun, adverb, preposition etc. which are used throughout in the document has to be removed to get proper result. For example 'Can listening be exhausting' so after removal of stop word can and be it will results in Listening, exhausting.

4.2.4 Noun filtering using Word net 3.0

Word similarity used the sizeable lexical database for English language to identify words for comparison. The version used in this study is Word net 3.0 [25, 30], which has 117,000 synonymous set, called synset. Word net has a path relationship between noun-noun and verb-verb only. This relationship is absent for other part of speech.

Example: A Voyage is a long journey on a ship or in a spacecraft.

In the above sentence Voyage, journey, Ship, and spacecraft are nouns.

4.2.5 Word overlap value calculation

After identifying the meaningful words, the overlap [2] values are calculated between the sentences in the document. Overlap value means that how much similar the words are in a sentence S1 and sentence S2. Similarly, the overlap value is calculated for sentence S3 and so on. Sum of all the overlap value represents the weight of the sentence and high value sentence is used in the summary. The calculated overlap values are arranged in decreasing order, and the first three highest values have been included in the summary between two documents.

4.2.6 Minimization of data using ontology

The ontology is used to minimize the summarized data further. Ontology of different domains i.e. Education, Sports, Technology, and Politics, etc. are stored in the database table name ontology. Ontology provides a common vocabulary of an area and defines, at different levels of formality, the meaning of terms and relationships between them. The relationship between token 1 to token 2 in a particular sentence is given by with 'is a' relation. The tokens are also matched with sentence S1 to sentence S2 to further minimizing the summarized data. Hence ontology tells about the relevance of the terms in a particular sentence to further summarize the data. The final summarized data is obtained after this step on which similarity calculation to be performed.

4.3 Similarity calculation of summarized data

N-grams token-based MD5 function, Winnowing fingerprint matching algorithm [26, 27] using Dice coefficient are used for similarity calculation of the summarized data. The steps are explained in brief as below.

4.3.1 N-gram formation

N-gram formation is a process of converting a string into substring. N is used for representing a number and tells how many words will be chosen in one gram. The input for N-gram is a preprocessed string, and the value of N. N can be 1, 2, …, n depending upon the user's requirement. If we take N = 1, the N-grams formed are known as unigrams. If we take N = 2, then the grams formed are bigrams. If we take N = 3, then grams formed are trigrams and so on. N-grams are the consecutive sequence of N character slice of a string. They can be evaluated using N = (p – m + 1), where p represents a number of letters in the document and m represents the size of N-grams. N-grams are generated from tokens after removal of spaces as shown below:

The size of N = 5.
Let a string ThisIsSKGram, 5-Grams derived from the string.

ThisI hisIs isIsS sIsSK IsSKG sSKGr SKGra KGram

4.3.2 Hash conversion

An ASCII value represents each character of gram. It converts grams into corresponding hash value [34]. Each character is converted to ASCII value. Hashing is a process of conversion of grams into short fixed-length value. It is performed because it is easy to find short length value than to find the original string. The search process will involve and then using it to find a match for a given value. Hashes are formed to avoid overwhelming computations. So we require a part of n grams to be used for comparison, and the n grams are converted to hash values.

The equation for hash formation can be given by

$$H(dk) = d1 * m^{(k-1)} + d2 * m^{(k-2)} + \cdots + d^{(k-1)} * m + d^k$$

where d: ASCII character, m: basis of primes, k: value of k-grams.

The input to hash function can be arbitrary, but the output is fixed referred to as n bit output. This process is the hashing of data. The converted small values are the hexadecimal value to be converted into decimal values. Several hashes are made for each document corresponding to each n-gram of the document.

4.3.3 Frame parsing

It is a method of converting hashes into frames. The input provided contains two parameters. The first parameter is the hash value, which is the output of the MD5 method. The second parameter is n, which tells the number of hashes to be kept in the frame. In this work, we have taken n as '4'. This parsing is done to ensure that minimum value is always available for selection from each frame. A function substring is used for providing the value of 'n'. The frames are created according to the size of n. The output is stored in an array list. Each frame will contain a value that will be used for comparison of two documents. Each frame will contain an equal number of hashes in it.

4.3.4 Process of fingerprint selection

In the previous phase, frames of equal size were formed. Each frame contains an equal number of hashes. For further processing of data, we need to choose a minimum value from each frame. All the values in a frame are compared to each other to find the least value. The reason for choosing the value as a minimum is that, the least value in one frame is likely to be the least value in other frames too. It said that a minimum of 'n' random number is smaller than one additional random number. The number of the values selected is much less as compared to the number of frames. It makes the document be represented by a small number of values and provides scalability to documents. When there are two similar hash values in two frames, then the value in the rightmost frame is chosen to be the least hash value. These all selected hash values together represent a document. This process uses the looping function to select the values on each window and array function to ensure that there is no similar value on the array as the result of fingerprint selection. The least hash values selected from Fig. 6 are 10 and 16.

4.4 The calculation process of document similarly

The Dice coefficient [35] of two sets is a measure of their intersection and it is scaled by their size (giving the value in the range 0 to 1). It is calculated as intersection over union of values.

$$Dice(X, Y) = 2|X \cap Y\ |/|X| + |Y|$$

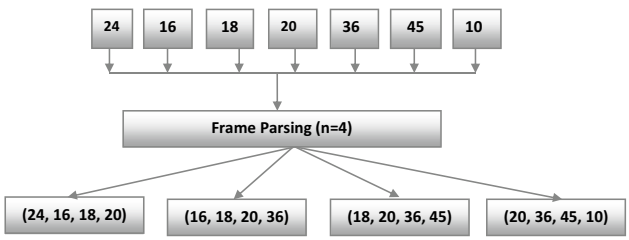


Fig. 6 Frame parsing

We take a string similarity measure the coefficient can be calculated for two strings X and Y as follow

X = night and Y = nauht, we would find the set of bigrams in each word

{ni, ig, gh, ht} and {na, au, uh, ht}

Each set has four elements, and the intersection of these two sets has only one element ht. Inserting, these numbers into the formula, we calculate

$$Similarity = \frac{2 * 1}{4 + 4} = 0.25$$

This similarity is compared with a pre-defined threshold if it is greater than the threshold it provides that the current web page is similar to the pages already in the database. This page will not add to the database.

5 Simulation set parameters

The proposed algorithm used similarity calculation that tells whether the new web page added into the database depends on a threshold value. By experimenting with similarity values calculation, the simulation setup parameter threshold set to be 0.65. If the similarity index is higher than this value, the web page does not add to the database because it already exists their; otherwise, it will be compared with other rows in the database. The page that is added to the database would be novel to other pages or documents into the database. In this way, the database will store only the novel pages at the crawling time, and search results always provide the novel results to the user query.

5.1 Set up parameters

The setup parameters are hardware, software and dataset used to perform the experiments. Table 2 show the setup parameters used to develop the overall experiments.

5.2 Performance parameters

To measure the efficacy of the proposed scheme several performance metrics are taken given under:

- *Redundancy removal (RR)* It is calculated as the difference between number of pages retrieved by the generic approach and number of pages retrieved by proposed approach.

  $$RR = abs\left(NP_{GA} - NO_{PA}\right)$$

  where, *RR* is the Redundancy Removal, NP_{GA} is the number of pages retrieved by the generic approach, and NP_{PA} is the number retrieved by the proposed approach
- *Memory overhead(MO)* For calculations of memory overhead number of pages retrieved by the generic approach and number of pages retrieved by proposed approach multiply by the page size are computed. This is the memory overhead used and is given by

  $$MO = NP_R * P_S$$

  where, *MO* is the memory overhead, and P_S is the page size in Megabytes.
- *Number of pages identified (NPI)* This gives the number of pages identified after the given search, which are relevant and novel. This is given as

Table 2 Setup parameters

Processor	Intel i3, 1.90 GHz processor	
Memory	RAM 4.00 GB, HDD 500 GB	
Software	Windows 10 Operating System, Microsoft Visual Studio 2012 (.NET) as front end, SQL server 2012 as a back end database.	
Data Set 1 (1634 documents)	Domain set 1	Sports
		Politics
		Education
		Technology
Data Set 2 (4430 documents)	Domain set 2	Health
		Entertainment
		Travel
		Zoology
Data Set 3 (4385 documents)	Domain set 3	Science
		World
		Business
		Transport

$NPI = NP_{PA}$

where, *NPI*, is the number of pages identified by the proposed approach, which is same as the pages retrieved by the proposed approach

6 Implementation

The implementation includes the Microsoft Visual Studio 2012 (.NET) as a front end and SQL server 2012 as a back end database. The SQL database includes three tables T_Categoty, T_website, and T_webpages. The table T_Category includes the website categories i.e., education, politics, sports, technology, health, entertainment, travel, and zoology. The T_Website and T_Webpages store the website related information together with web pages related information. The database stores the URL of the query together with its HTML tags, metadata tags, etc. It also includes the table ontology, Senti Dictionary table (Word Net 3.0), and overlap table to store ontology together with overlap calculation information. The dictionary contains the URLs used by the user to search any query word.

Figure 7 is a Search Engine interface that appears when the user clicks on the search button. This interface includes keyword to be search based on the advanced search and field-specific search.

Figure 8 showed the search results when the user typed the keyword or topic on the search interface to the generic crawler. It shows the results for the keyword 'code', which already stored in the database for the technology category. This search result is showing the redundant and relevant webpage for the given query, which a tedious and time-consuming task for the user to read whole documents. The proposed methodology included the text summarization, syntactic similarity, plus semantic similarity to overcome the limitations of the generic crawler. This work provides the relevant and novel results to the user's query and filters out the redundant ones.

When the same query as in Fig. 8 runs on the proposed crawler interface as in Fig. 9, it filters out redundant ones and displays only the relevant and novel pages.

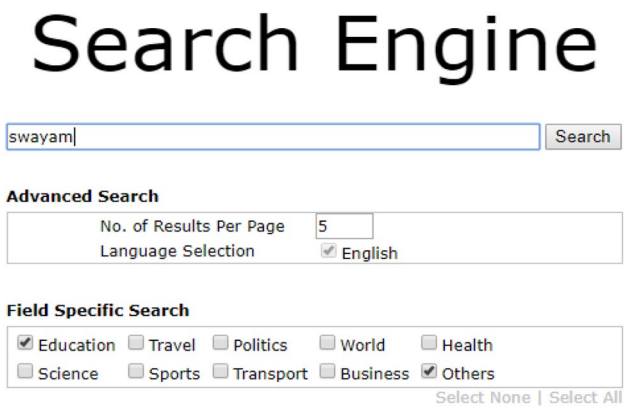
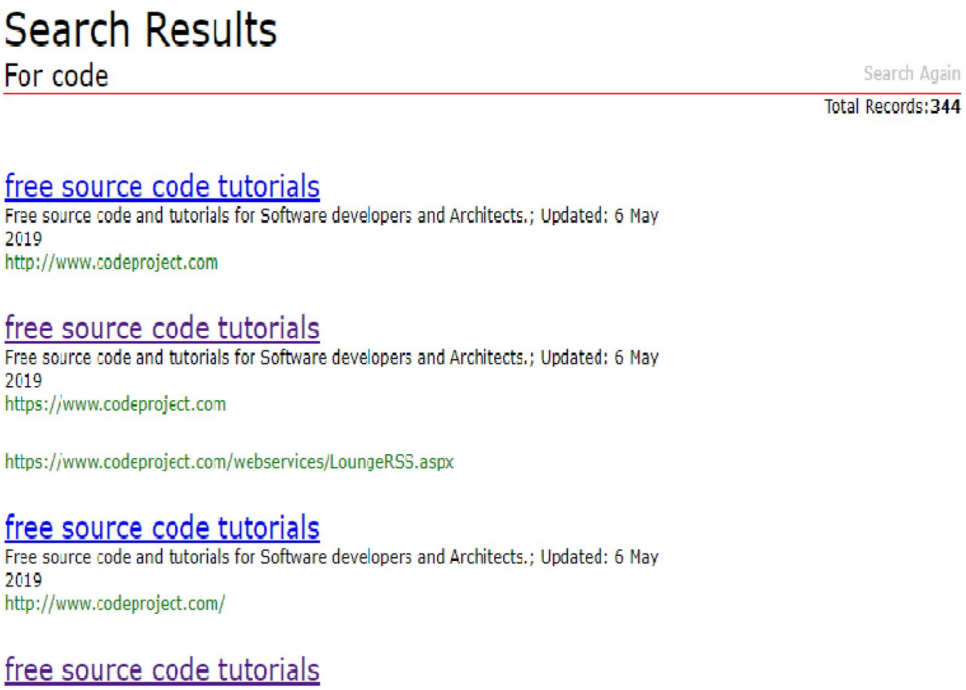


Fig. 7 Search engine interface

Fig. 8 List of webpages for the query 'code' on generic crawler search interface



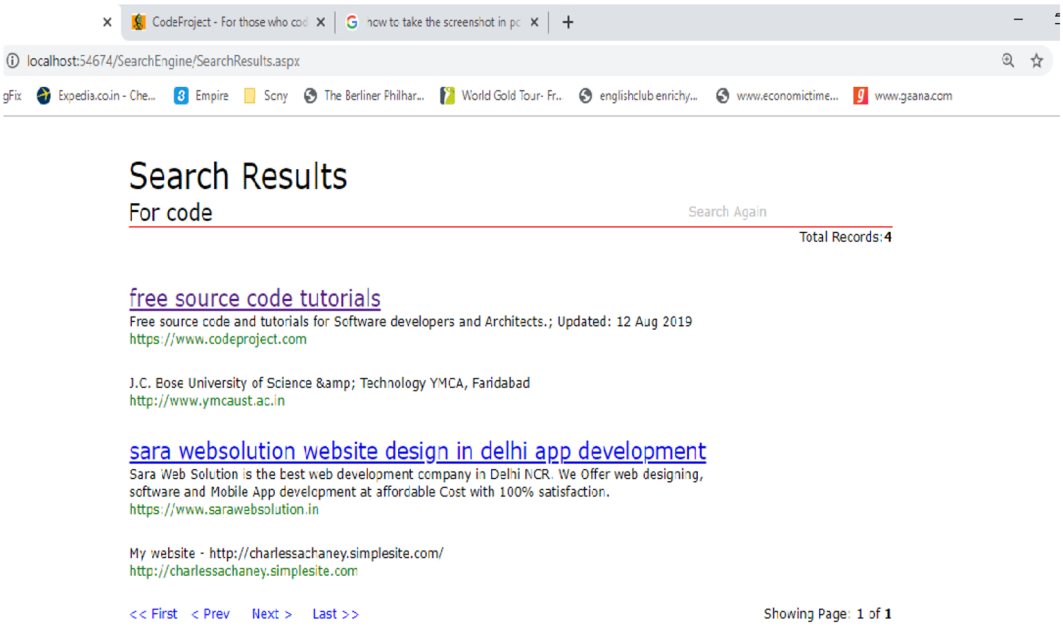


Fig. 9 List of webpages for the query 'code' on proposed crawler search interface

Table 3 Comparison of generic crawler and proposed crawler novelty

Domain	Query	Generic crawler (no. of pages retrieved)	Proposed crawler (no. of novel pages retrieved)	Redundant pages
Sports	Sports	80	5	75
	Ball	70	3	72
	Boxing	40	2	38
	Cycling	30	1	29
Politics	Politics	100	4	96
	Election	60	3	57
	Campaign	40	2	38
	Leadership	40	2	38
Education	Education	170	9	161
	YMCA	50	1	49
	University	110	5	105
	Board	140	6	132
Technology	Code	344	4	340
	Web	130	8	122
	HTML	130	8	122
	Java	120	6	114
	Total	1654	69	1588

7 Results and discussion

Users can press on the search button after typing any topic or keyword on the search interface with the number of pages to be displayed together with ticking on the field-specific search i.e. education, sports, and politics, etc.

7.1 Data Set 1

Table 3 shows the different queries that executed for different domains on the Search Engine Interface of the generic crawler and proposed crawler novelty. The results of these queries are stored in the crawler indexed database of the generic method and proposed method.

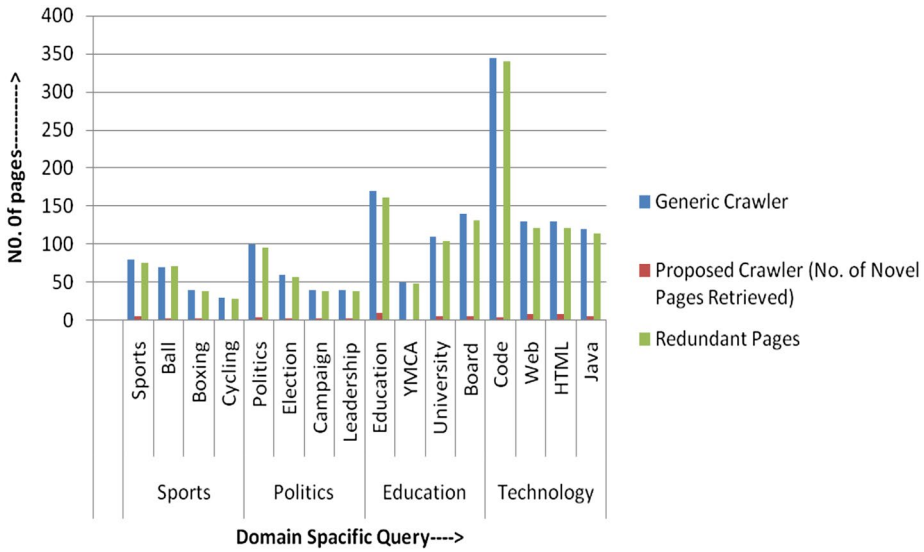


Fig. 10 Comparison of generic crawler and proposed crawler novelty results

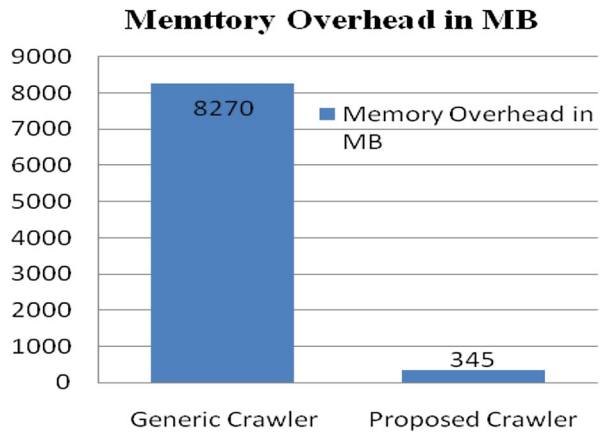


Fig. 11 Comparison of memory overhead

Result analysis 1 As shown in Table 3 and Fig. 10 above that Generic crawler search provide 80, 70, 40 and 30 documents for queries '*sports*', '*Ball*', '*boxing*', and '*cycling*' under the domain '*Sports*', respectively. On the other hand, the proposed approach provides 5, 3, 2, and 1, which all are novel and filter out the remaining ones.

Result analysis 2 As shown in Table 3 and Fig. 10 above that Generic crawler search provide 100, 60, 40 and 40 documents for queries '*politics*', '*election*', '*campaign*', and '*leadership*' under the domain '*Politics*', respectively. On the other hand, the proposed approach provides 4, 3, 2, and 2, which all are novel and filter out the remaining ones.

Result analysis 3 As shown in Table 3 and Fig. 10 above that Generic crawler search provide 170, 110, 50 and 140 documents for queries' *education*', '*ymca*', '*university*', and '*board*' under the domain '*Education*', respectively. On the other hand, the proposed approach provides 9, 1, 5, and 6, which all are novel and filter out the remaining ones.

Result analysis 4 As shown in Table 3 and Fig. 10 above that Generic crawler search provide 344, 130, 130 and 120 documents for queries' *code*', '*web*', 'html', and '*java*' under the domain '*Technology*', respectively. On the other hand, the proposed approach provides 4, 8, 8, and 6, which all are novel and filter out the remaining ones.

Memory overhead As shown in Fig. 11, If a page of size is 5 MB, then the generic approach memory requirement is 1654*5 = 8270 MB, and according to the proposed approach, the memory requirement is 69*5 = 345 MB.

7.2 Data Set 2

Table 4 shows the different queries that executed for different domains on the Search Engine Interface of the generic crawler and proposed crawler novelty. The results of these queries are stored in the crawler indexed database of the generic method and proposed method.

Result analysis 1 As shown in Table 4 and Fig. 12 above, that Generic crawler search provides 220, 300, 190 and 390 documents for queries' patient', 'medicine', 'doctor', and 'health' under the domain name 'Health', respectively. On the other hand, the proposed approach provides 40, 50, 30, and 70, which all are novel and filter out the redundant pages.

Result analysis 2 As shown in Table 4 and Fig. 12 above that Generic crawler search provide 120, 100, 90 and 310 documents for queries '*movie*', '*music*', '*comedy*', and '*entertainment*' under the domain name '*Entertainment*', respectively. On the other hand proposed approach provide 20, 20, 20 and 60 which all are novel and filter out the redundant pages.

Result analysis 3 As shown in Table 4 and Fig. 12 above that Generic crawler search provide 310, 220, 500 and 390 documents for queries '*tourism*', '*travel*', '*tour*', and '*holiday*'

Table 4 Comparison of generic crawler and proposed crawler novelty

Domain	Query	Generic crawler (no. of pages retrieved)	Proposed crawler (no. of novel pages retrieved)	Redundant pages
Health	Patient	220	40	180
	Medicine	300	50	250
	Doctor	190	30	160
	Health	390	70	320
Entertainment	Movie	120	20	100
	Music	100	20	80
	Comedy	90	20	70
	Entertainment	310	60	250
Travel	Tourism	310	50	260
	Travel	500	80	420
	Tour	220	80	140
	Holiday	390	60	330
Zoology	Cryptozoology	50	10	40
	Biology	460	60	400
	Zoology	420	60	360
	Animal	360	50	310
	Total	4430	760	3670

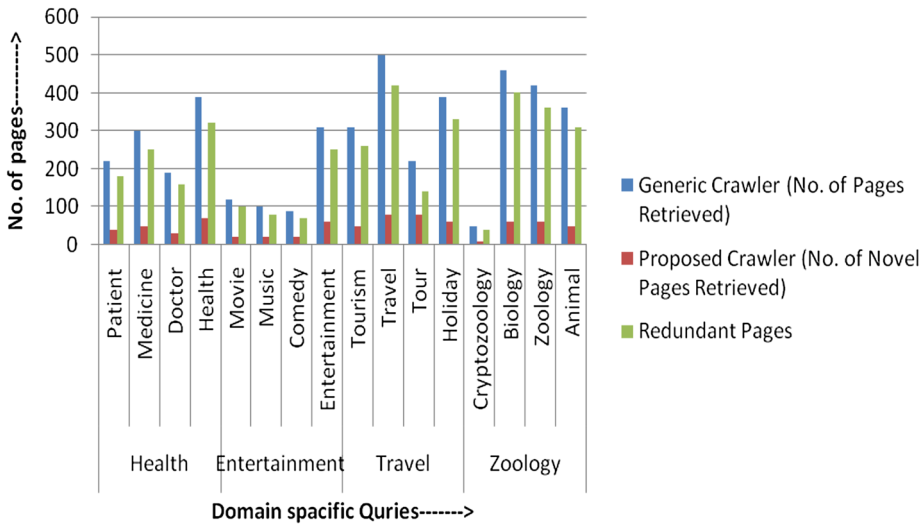


Fig. 12 Comparison of generic crawler and proposed crawler novelty results

under the domain name '*Travel*', respectively. On the other hand, the proposed approach provides 50, 80, 80, and 60, which all are novel and filter out the redundant pages.

Result analysis 4 As shown in Table 4 and Fig. 12 above that Generic crawler search provide 50, 460, 420 and 360 documents for queries' *cryptozoology*', '*biology*', '*zoology*', and '*animal*' under the domain name '*Zoology*', respectively. On the other hand, the proposed approach provides 10, 60, 60, and 50, which all are novel and filter out the redundant pages.

Memory overhead As shown in Fig. 13, if a page of size is 5 MB, then the generic approach memory overhead is 4430*5 = 22,150 MB, and according to the proposed approach, the memory overhead is 760*5 = 3800 MB.

7.3 Data Set 3

Table 5 shows the different queries that executed for different domains on the Search Engine Interface of the generic crawler and proposed crawler novelty. The results of these

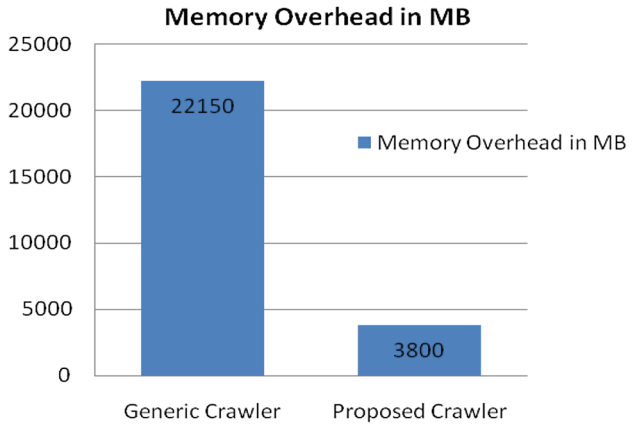


Fig. 13 Comparison of memory overhead

queries are stored in the crawler indexed database of the generic method and proposed method.

Result analysis 1 As shown in Table 5 and Fig. 14 above, that Generic crawler search provided 130, 100, 80 and 160 documents for queries '*Geophysics*', '*Scientist*', '*Laboratory*', and 'Laws' under the domain name '*Science*', respectively. On the other hand, the proposed approach provided 15, 20, 10, and 20, which all are novel and filter out the redundant pages.

Result analysis 2 As shown in Table 5 and Fig. 14 above that Generic crawler search provided 390, 360, 310 and 320 documents for queries '*universe*', '*nature*', '*society*', and '*people*' under the domain name '*World*', respectively. On the other hand proposed approach provide 45, 38, 20 and 28 which all are novel and filter out the redundant pages.

Result analysis 3 As shown in Table 5 and Fig. 14 above that Generic crawler search provided 280, 220, 180, and 170 documents for queries '*service*', '*merchandise*', '*manufacturing*', and '*partnership*' under the domain name '*Business*', respectively. On the other hand, the proposed approach provides 241, 196, 150, and 145, which all are novel and filter out the redundant pages.

Result analysis 4 As shown in Table 5 and Fig. 14 above that Generic crawler search provided 480, 430, 415 and 360 documents for queries '*bus*', '*car*', '*truck*', and '*vehicle*' under the domain name '*Transport*'. On the other hand, the proposed approach provides 50, 40, 30, and 42, which all are novel and filter out the redundant pages.

Memory overhead As shown in Fig. 15, if a page of size is 5 MB, then the generic approach memory overhead is 4385*5 = 21,925 MB, and according to the proposed approach, the memory overhead is 478*5 = 2390 MB.

From the above results, it has been cleared that this proposed approach provided the novel documents for a given

Table 5 Comparison of generic crawler and proposed crawler novelty

Domain	Query	Generic crawler (no. of pages retrieved)	Proposed crawler (no. of novel pages retrieved)	Redundant pages
Science	Geophysics	130	15	115
	Scientist	100	20	80
	Laboratory	80	10	70
	Laws	160	22	138
World	Universe	390	45	345
	Nature	360	38	322
	Society	310	20	290
	People	320	28	292
Business	Service	280	39	241
	Merchandise	220	24	196
	Manufacturing	180	30	150
	Partnership	170	25	145
Transport	Bus	480	50	430
	Car	430	40	390
	Truck	415	30	385
	Vehicle	360	42	318
	Total	4385	478	3907

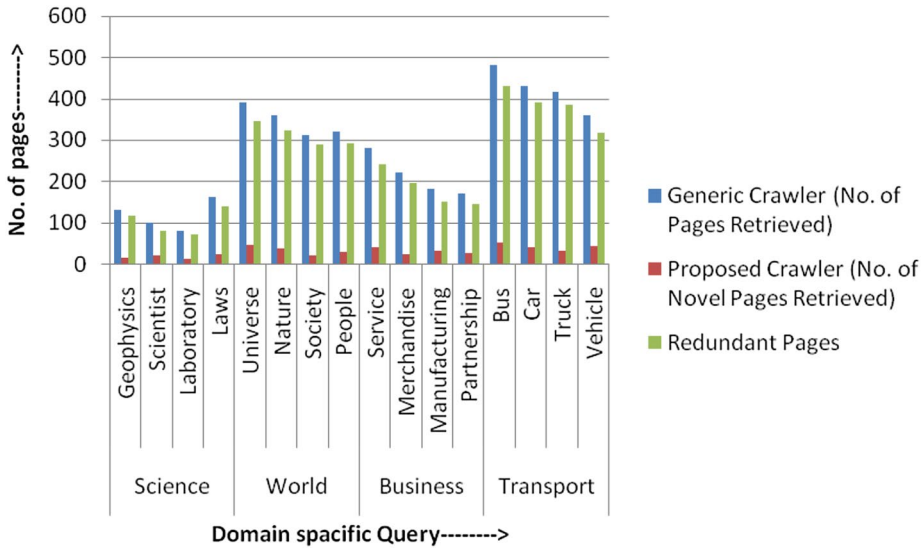


Fig. 14 Comparison of generic crawler and proposed crawler novelty results

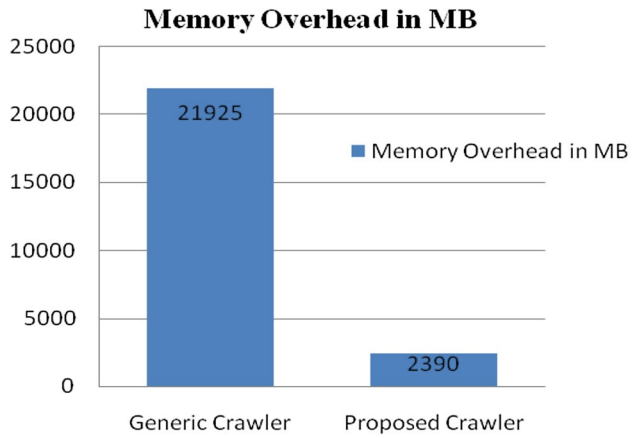


Fig. 15 Comparison of memory overhead

query with minimum memory overhead and filtered out the redundant documents.

8 Conclusion

In this paper a novel technique based on extractive text summarization using ontology, semantic similarity using word net 3.0, and similarity calculation using winnowing algorithm is proposed. The result after comparison with generic crawler present following inferences:

- After performing experiments with keywords/query words from different domains the proposed work gives least redundant results. The average redundancy is reduced to 88% of all the results.
- Reduced redundancy provides novel results for the prescribe search rather than replicating the previous results. This results in effective search effort.
- Memory requirement for the search results also reduce to large extent.
- One of the, main feature of this technique is that number of pages identified after the given search are very less as compared to generic technique. This results in the elimination of repeated occurrence and less memory requirement with less execution time.

Hence, it is concluded that this proposed approach can be used successfully in the field of information retrieval.

Compliance with ethical standards

Conflicts of interest The authors declare they have no conflict of interest.

References

1. Brants T, Chen F, Farahat A (2003) A system for new event detection. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 330–337
2. Gupta SB (2012) The issues and challenges with the web crawlers. Int J Inf Technol Syst 1(1):1–10
3. Zhao L, Zhang M, Ma S (2006) The nature of novelty detection. Inf Retr 9(5):521–541
4. Zhang Y, Callan J, Callan J, Minka T (2002) Novelty and redundancy detection in adaptive filtering. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 81–88
5. Allan J, Gupta R, Khandelwal V (2001) Temporal summaries of new topics. In: Proceedings of the 24th annual international

ACM SIGIR conference on research and development in information retrieval, pp 10–18
6. Allan J, Wade C, Bolivar A (2003) Retrieval and novelty detection at the sentence level. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 314–321
7. Li X, Croft WB (2005) Novelty detection based on sentence-level patterns. In: Proceedings of the 14th ACM international conference on information and knowledge management. ACM, pp 744–751
8. Kwee AT, Tsai FS, Tang W (2009) Sentence-level novelty detection in English and Malay. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, Berlin, Heidelberg, pp 40–51
9. Bentivogli L, Clark P, Dagan I, Giampiccolo D (2011). The seventh PASCAL recognizing textual entailment challenge. In: TAC
10. Soboroff I, Harman D (2005) Novelty detection: the TREC experience. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp 105–112
11. Stokes N, Carthy J (2001) Combining semantic and syntactic document classifiers to improve first story detection. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, pp 424–425
12. Franz M, Ittycheriah A, McCarley JS, Ward T (2001) First story detection, combining similarity and novelty based approach. In: Topic detection and tracking (TDT) workshop report
13. Larkey LS, Allan J, Connell ME, Bolivar A, Wade C (2002) UMass at TREC 2002: cross language and novelty tracks. Massachusetts Univ Amherst Center for Intelligent Information Retrieval
14. Tsai MF, Hsu MH, Chen HH (2003) Approach of information retrieval with reference corpus to novelty detection. In: TREC, pp 474–479
15. Alqaraleh S (2011) Elimination of repeated occurrences in image search engines. Technical report, Eastern Mediterranean University, North Cyprus
16. Alqaraleh S, Ramadan O (2014) Elimination of repeated occurrences in multimedia search engines. Int Arab J Inf Technol 11(2):134–139
17. Sravanthi P, Srinivasu B (2017) Semantic similarity between sentences. Int Res J Eng Technol (IRJET) 4(1):156–161
18. Karkali M, Rousseau F, Ntoulas A, Vazirgiannis M (2013) Efficient online novelty detection in news streams. In: International conference on web information systems engineering. Springer, Berlin, Heidelberg, pp 57–71
19. Dasgupta T, Dey L (2016) Automatic scoring for innovativeness of textual ideas. In: Workshops at the thirtieth AAAI conference on artificial intelligence
20. Ghosal T, Salam A, Tiwari S, Ekbal A, Bhattacharyya P (2018) TAP-DLND 1.0: a corpus for document level novelty detection. arXiv preprint arXiv:1802.06950
21. Lee CS, Kao YF, Kuo YH, Wang MH (2007) Automated ontology construction for unstructured text documents. Data Knowl Eng 60(3):547–566
22. Henze N, Dolog P, Nejdl W (2004) Reasoning and ontologies for personalized e-learning in the semantic web. J Educ Technol Soc 7(4):82–97
23. Hovy E, Lin CY (1999) Automated text summarization in SUMMARIST. Adv Autom Text Summ 14:81–97
24. Simmons S, Estes Z (2006) Using latent semantic analysis to estimate similarity. In: Proceedings of the cognitive science society, pp 2169–2173
25. Leacock C, Chodorow M (1998) Combining local context and WordNet similarity for word sense identification. WordNet Electron Lex Database 49(2):265–283
26. Wibowo AT, Sudarmadi KW, Barmawi AM (2013) Comparison between fingerprint and winnowing algorithm to detect plagiarism fraud on Bahasa Indonesia documents. In: 2013 international conference of information and communication technology (ICoICT). IEEE, pp 128–133
27. Alzahrani SM (2009) Plagiarism auto-detection in arabic scripts using statement-based fingerprints matching and fuzzy-set information retrieval. Doctoral dissertation, Universiti Teknologi Malaysia
28. Meng L, Huang R, Gu J (2013) A review of semantic similarity measures in wordnet. Int J Hybrid Inf Technol 6(1):1–12
29. Ceska Z, Hanak I, Tesar R (2007) Teraman: a tool for N-gram extraction from large datasets. In: 2007 IEEE international conference on intelligent computer communication and processing. IEEE, pp 209–216
30. Hussein AS (2015) Arabic document similarity analysis using n-grams and singular value decomposition. In: 2015 IEEE 9th international conference on research challenges in information science (RCIS). IEEE, pp 445–455
31. Pooja KS, Bhatia KK (2019) Hashing and clustering-based novelty detection. SSRG Int J Comput Sci Eng 6(6):121–126
32. Jiayi P, Cheng CPJ, Lau GT, Law KH (2008) Utilizing statistical semantic similarity techniques for ontology mapping—with applications to AEC standard models. Tsinghua Sci Technol 13(S1):217–222
33. Rus V, Lintean M, Banjade R, Niraula N, Stefanescu D (2013) Seminar: the semantic similarity toolkit. In: Proceedings of the 51st annual meeting of the association for computational linguistics: system demonstrations, pp 163–168
34. Cheddad A, Condell J, Curran K, McKevitt P (2010) A hash-based image encryption algorithm. Opt Commun 283(6):879–893
35. Meadow CT, Boyce BR, Kraft DH (1992) Text information retrieval systems, vol 20. Academic Press, San Diego