Int. j. inf. tecnol. (May 2022) 14(3):1211–1219
https://doi.org/10.1007/s41870-020-00422-y

ORIGINAL RESEARCH

Web URLs retrieval with least execution time using MPV clustering approach

Sunita[1] · Vijay Rana[2]

Received: 23 February 2019 / Accepted: 10 January 2020 / Published online: 18 January 2020

Abstract Web content searching in the advanced time requires the disclosure of examples from user queries, yet that too at high speeds. In different examples and learning extraction systems utilized that attempts to find information from the hyperlink designs. Removing such data devours time and also, recorded contents through web search need particularity. The proposed methodology is ordered into four parameters, for example, First is watchword division, second is information getting, the third one is execution time and the fourth one is the number of URLs gotten in a question string. Watchword division goes about as expel stop words, stemming, spell check, tokenization from the specific user question. Information getting is prepared to mining watchwords from a user string coordinated inside the information word reference. The execution time depends on getting the right watchwords from the user question string and mine the number of URLs coordinated with user catchphrases. However, the significance of catch phrases in user queries. The separated watchwords which decide through centrality measure division stage are tried to decide important URLs legitimize the more solid outcome. This paper proposed a novel regulated most probable learning component utilized for exhibitions appeared regarding execution time, data extraction, number of URLs got in web search. The consequences of the proposed framework are contrasted and approved and existing clustering models with most probable value clustering (MPV).

Keywords Preprocessing · Keyword segmentation · Data fetching

1 Introduction

Data getting is an approach to comprehend the web data and discovering explicit and critical data from web data utilizing data extraction procedures [1]. The data is brought from unstructured web data asset and various data configurations like tables, HTML labels, and so on.

The representation of user question contents characterized as a lot of words must be tokenized so as to remove significant watchwords. In the data recovery (IR), watchwords are generally used to decide user inquiry [2]. The importance of watchwords from the user inquiry is called catchphrase extraction. The uses of watchwords like synopsis, arrangement, location, separating, and clustering. Catchphrase extraction [3] is a convoluted assignment whenever performed physically, so it is essential to have a viable programmed watchword bringing process. For making a mechanized catchphrase getting process two primary advances are pursued: right off the bat content is pre-prepared for making it appropriate for learning calculation and after that learning calculation is connected to the pre-handled data for sensing.

For recovery of a catchphrase, different levels like to stop-words evacuation, tokenization, stemming and spell check are utilized, in the preprocessing stages.

✉ Sunita
sunitamahajan2603@gmail.com

Vijay Rana
vijay.rana93@gmail.com

[1] Department of Computer Science, Arni University, Kangra, India

[2] Department of Computer Science, Sant Baba Bhag Singh University, Khiala, India

1.1 Preprocessing phases

Preprocessing [4] is the most important part of data cleaning. It can remove the noise from the unstructured data. There are some phases are used in the preprocessing shown as below (Fig. 1).

- Stop word removal
- Tokenization
- Stemming
- Spell checks

1.1.1 Stop word removal

The principal level, stop word [5] evacuation in which pointless words expelled from the content since it makes the content look heavier and it's very little critical to an expert. The dimensionality of term space is diminished utilizing the expulsion of stop words [6]. The regular stop words that have no importance in user question are an article, relational words, and pronouns, and so on. Since these words are not considered as a catchphrase so it ought to be evacuated. The different systems that are used as given beneath:

- The classic method
- Z-method
- MI (mutual–information) method
- TBRS (Term based random sampling)

The classic method: it will remove stop words that gathered from a pre-compiled list.

- *Z-method* It utilizes Zipf's law that removed stop words, including words whose frequency is high and the words which occur only once. Also, it's considered low inverse document frequency for removing words.
- *MI (Mutual–Information) method* This method considers mutual information between the given term and document class. It gives suggestion on the basis of information given by the term.
- *TBRS (Term based random sampling)* It manually detects the stop words and it works on randomly selecting data from separate chunks. After that ranking is given to those chunks that are gathered through random selection.

1.1.2 Tokenization

The significant components like words, images, and expressions in a flood of content are known as a token [7]. The way toward breaking tokens from the flood of content is called tokenization. It investigates words from the user inquiry to distinguish important data [8]. This rundown of tokens has given as contribution for further handling. It is valuable for giving intelligibility of the record or inquiry. The different strategies that are utilized for tokenization are as given underneath:

- *Nlpdotnettokenizer* It is based on a neural network that performs tokenization. It is used innatural language processing tasks.
- Mila Tokenizer: It divides input text into tokens using the XML format.
- NLTK tokenizer: It is an important toolkit that utilizes a python program for performing tokenization. It provides text processing libraries and easy to use interface.
- MBSP word tokenizer: It is based on memory-based learning and used induction of linguistic knowledge for tokenization and sentence splitting.

1.1.3 Stemming

The extraction of subparts from the given word is known stemming. For instance: bolstered, underpins, supporting can be gushed to the "support". It will dispense with addition and prefix of a word from the term [9]. In stemming word's morphological structure is changed over into its stem by expecting that it semantically has a similar importance. For utilizing stemmer two central matters are considered:

- Morphological forms
- Words

The words that have similar base importance are considered as morphological structure and it is changed to stem. Extra words that are not mapped kept isolated. The accompanying procedures are used for stemming:

1. Truncating algorithms
2. Statistical algorithms

- *Truncating Algorithms* These algorithms are utilized to evacuate postfix and prefix of a word, it truncates

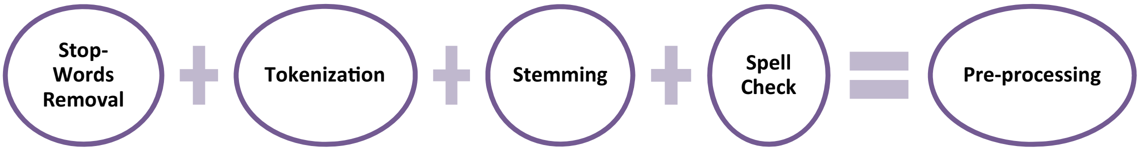

Fig. 1 Producer of user query preprocessing

the math image from a word. It implies in these calculations the words without any letters are held and evacuate the remainder of the letter. The accompanying stemmer used truncating calculations

- Lovins stemmer
- Porters stemmer
- Husk stemmer
- Dawson stemmer
- *Statistical algorithms* These are stemmer that utilizes statistical techniques and analysis for stemming. The following stemmer uses statistical algorithms:

1. N-gram stemmer
2. HMM stemmer
3. YASS stemmer

- *N-gram Stemmer* The similarities in a string are used to convert the inflation of word tostem, in this method. In this *n* consecutive characters are extracted from a word that is adjacent. It is language independent.
- *HMM stemmer* This method is based on finite automata that utilizing probability function.In this, the probability is computed for each path and then a most significant probable path is considered that create automata. In this string is considered that consist of a sequence of letters and then the concatenation of subsequences are found that provide a result.
- *YASS stemmer* this method creates a group of letters using distance measures and a hierarchical approach. The result of clustering than considers for removing stems in words.

1.1.4 Spell checks

Subsequent to stemming the words are considered for checking linguistic slip-ups. For this reason, spell checking is finished utilizing the word reference. The lexicon comprises of linguistically right words. In the event that there is a mix-up in words than it is naturally adjusted utilizing a lexicon that is created disconnected.

2 Literature survey

In rank-based algorithm is proposed [6]- [10] for semantic web search. In this algorithm, the criteria based information derived from the huge semantic environment and then user query are analyzed. It mainly utilized page relevance and then provides a relevance score for a web page. The page relevance measure involved graph-based representations along with the probability aware approach. The results show that cost reduction and accuracy is better. But it does not base on web repositories and multiple ontologies are also not used.

In spam based web search technologies [5–7] are used and it is used to detect web spam. It firstly detects the content features than non-spam pages are identified. After that spam pages that are made by spammers are detected. The results show that it has helped in spam detection. But it does not focus on the semantics which is utilized in searching and also there are no appropriate methods used for detection content features.

In proposing a weighted page rank algorithm [11] in a mobile system that is used to link the structure of various web pages and calculate the rank of the pages. If a page has more outgoing links than it has the highest rank. This rank is used to give probability about the particular page when a user query is given. In this algorithm, the current rank of the page is utilized for estimating the probability.

In describing an algorithm [12] that analytic structure of web page links and the authority is provided in that link. - Then, according to the user query information is stored in authority pages. It works in two steps, firstly sampling is to be done than iterative calculation is performed to solve the user query. It calculates the rank of pages.

In descriptive ontology-based techniques [13] that are based on index and relationships. This provides a better search and also provide pages that are based on the user interest. It also utilizes semantics for searching the web pages according to a user query. It only displays those pages that achieved the relationship to the user query.

In describing a system that utilizes the OWL technique [14] for semantic illustration and utilized for monitoring use. In this data recuperation method is used for highlighting the user interest and then semantic comparability is tested. It gives the group estimation that highlight data which is according to a user query.

In describing intelligent web [13] service that uses ontology and retrieves the information in a précised manner. It provides an intelligent agent that analysis, user query and gives data related to it according to most searches. It utilizes mining and shortlists the web pages that are semantically related to it. It decreased irrelevant search result and précised knowledge discovery is made.

The techniques being analyzed does not include the clustering that is used in the proposed system. This lack of cluster usage leads to high execution time in browsing with an extended search space. The proposed system is described in the next section.

3 Objectives/aim

From the literature survey, it is extracted that work has been done towards pre-processing of user query, but least amount of work is done towards location based clustering mechanism that emphasizes on pre-processing phase for keyword extraction and word processing and classification of user query includes location sensitive site extraction procedure. The aim and objective of this study to achieve the following objectives:

- To fetch keywords from user query during the preprocessing.
- To understand and maintain a database of related URLs alternative to the keywords.
- To apply MPV clustering approaches to generate groups for appropriate URLs against user keywords.

The aim of this study provides precise and accurate information to the users. And reduce the web search time and eliminate the memory space in web searching.

4 Proposed methodology

In this section our proposed system computes the meaningful keywords after that preprocessing. The proposed system as shown Fig. 2 for detecting and controlling URLs. The URLs comprises with user keywords. In the proposed system firstly keywords segmentation is responsible to detect keywords. The collected keywords data transfer to match with URLs database. After matching process to fetch meaningful URLs to improve the execution time to detect in the term number of URLs. The purposes of accurate information fetching apply MPV clustering. The entire work of the proposed system is categorized into phases.

5 Phases of simulation

5.1 Phase 1: keyword segmentation

In this phase perform various preprocessing steps on user query to determine the keywords. This phase is called by keyword segmentation.

5.2 Identifying tokens from the user query

This phase consists of extracting the meaningful work which exists in the dictionary representing meaningful words. All these words are known as tokens. In addition to identifying tokens, it also identifies the misspelled words and suggests corrections. The corrected words are replaced with the existing URL words.

6 Removing stop words

Stop word removal becomes a need of the hour to reduce the time required to perform searches. These words are bound to ignore by the search engine. These words will be removed in the proposed mechanism. To accomplish this, stop words dictionary is maintained. Once the stop words are removed a length of query string subsequently reduced.

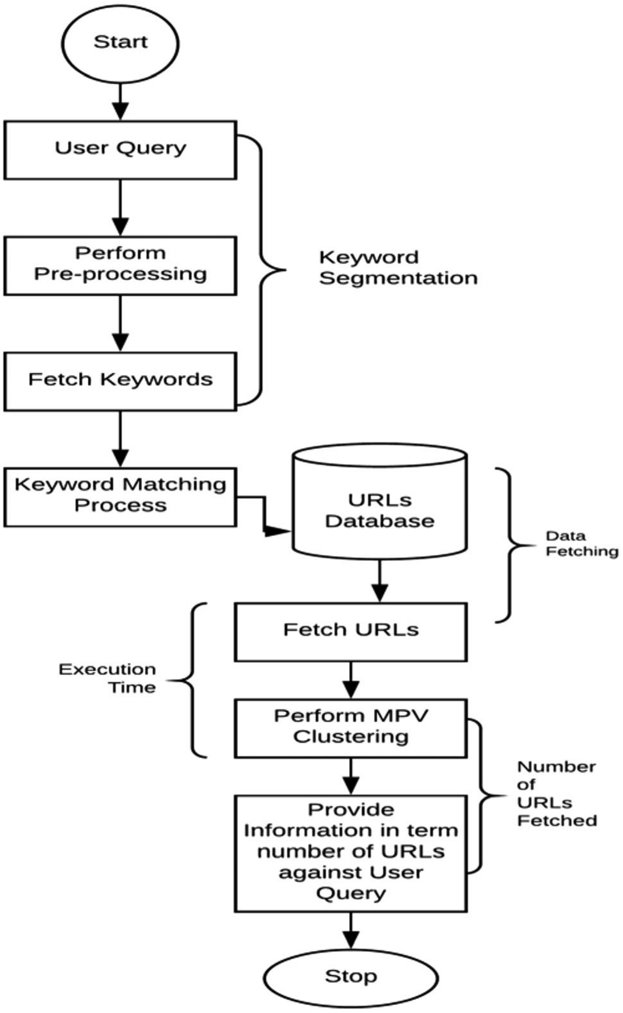

Fig. 2 Process of URLs fetching

7 Phase 2: data fetching

7.1 Extracting keywords

Keywords in web searching represent the most profound search words. The frequency of occurrence of keywords in URL is high as compared to a normal word. Thus keywords identification in the proposed system is achieved using a statistical measure known as a model. The highest frequency of the word will be directly proportional to the probability of keywords.

7.2 Phase 3: execution time

This phase is the main phase, in the proposed model it can be analyzed the execution time to retrieve the URLs against user query by using the MPV clustering approach.

8 Forming clusters

This phase presents the distinguished from other browsers. In this phase MPV (most probable values) clustering mechanism is proposed. This is a simple mechanism in which keywords extracted from the query string are stored within the dataset. These keywords are accompanied by the count variable. This variable increases as the same keywords appear again within a string URL. Euclidean distance is evaluated corresponding to each keyword. Threshold distance is also maintained. In case Euclidean distance is less than the threshold distance then keywords are collected within the cluster.

8.1 Phase 4: URLs fetch

In this phase proposed model determined by perform MPV clustering toextract more precise URLs provide

```
Algorithm MPV_Cluster
Start
            URL represents the user input which is stored within 'U' variable
            'db' indicates the dictionary of meaningful words
Extracted_Tokens indicates the meaningful tokens
Extracted_stop is the variable for maintaining Query without stop words
            Stop is the database for stop words
            MPV is the historical information of keywords searched
            Cluster is the group of keywords

    a)  Read the URL from the query string
        U=URL(Query_String)
    Phase to extract tokens from the query string
    b)  For i=1: length(U)
            For j=1: length(db)
        If(U(i)==db(j))
        Extracted_Tokens=Extracted_Token+" "+U(i)
                  Break
                 End of if
          End of for
        End of for
        Phase to remove stop words and extracting keywords
    c)  Extracted_stop= Extracted_Tokens
    d)  For i=1: length(Extracted_stop)
          For j=1: length(stop)
        If(Extracted_stop(i)!=stop(j))
        Without_stop= Without_stop+" "+Extracted_stop(i)
            End of if
         End of for
        End of for
    e)  Phase to find most probable clustering
        For i=1:length(Without_stop)
           For j=1: length(MPV)
             If Without_stop(i)==MPV(j)
        $Count_i=Count_i+1$
             End of if
           End of for
          End of for
    f)  finding distance in terms of count and storing the result with corresponding cluster index
        For i=1: length(Without_stop)
        If(count($Without\_stop_i$)<=Threshold)
        $Cluster_i$= $Without\_stop_i$
           End of if
        End of for
    g)  Retrieve website URLs corresponds to $Without\_stop_i$ and print result in terms of execution time
        Stop
```

information in term number of URLs against User. The simulation setup is given below section.

9 Algorithm

The algorithm for the proposed system is given as under.

Next section gives the performance analysis and results corresponding to the algorithm given above.

9.1 Simulation setup

The simulation setup consists of software and hardware requirements critical for proposed browsing scheme. The primary setup requires ASP.net to perform simulation of basic browsing mechanism. The dictionary specifies the stop word is contained within MS-ACCESS database having a.mdb extension. WordNet is employed for checking stop words against the user query string. Stop word elimination after the first phase causes the generation of keywords. The validity of this approach is tested against multiple distributed system. In addition, internet setup is required for the testing purpose. The configuration of the system used for the proposed system is given in Table 1.

Software configuration for the proposed system uses ASP.net for server-side scripting and MS-Access is used to provide a caching mechanism. Table 2 gives software configuration requires for a proposed system.

Table 2 Software requirements for proposed system

Software	Configuration/version
ASP.NET	2013
MS-ACCESS	2013
WordNet	3.1

10 Performance analysis and discussion

The results obtained through simulation are discussed in this section. A query string used in the proposed work is represented by ‘Q’, an actual user query is a collection of strings. Phases correspond to propose work applied to the strings give result in modules. The first phase corresponds to pre-processing that is subdivided into tokenization, stemming and stop word elimination and the result corresponds to this phase is given in Table 1.

This phase, although additional time in dissimilar word handling, but it yields a better result in terms of meaningful URL fetching. User query initially is passed through tokenization where an entire query is parsed and compared against the token dictionary. The process yields meaningful and dissimilar words. The query is then formed again from the tokens. The tokens, then compared against the stop word dictionary to eliminate them from a user query. Once stop words eliminated from the string, spell checking phase checks the words and propose corrections. In case a user accepts the corrections, words in strings are replaced. This will conclude the preprocessing phase. Keyword extraction is critical since meaning information processing is achieved only if keyword extraction is successful. Keyword extraction phase compares the extracted words after the correction phase. Correction phase gives the optimal result and keyword extraction phase consume less time since the correction is not a part of an extraction. This distinguishment of keyword extraction and correction enhances performance in terms of time consumed in browsing. The result corresponding to keyword extraction is highlighted in Tables 2.

The parametric results of pre-processing and keyword extraction phase without clustering when browsing is performed by the user is listed in Table 3. Execution time parameter indicates the total time it takes to produce the result in the form of maximum possible URLs. Although limited websites are used for the purpose simulation but still result of time consumption is less than 2 s for each user query (Tables 4 and 5) (Fig. 3).

The result produced by the proposed mechanism performs order by determining the most frequent keywords searched. This will setup locality of reference to enhance the speed with which searching operation is being performed. The rank-based mechanism is termed as most probable clusters and during searching, an only relevant cluster is required to be searched. The result corresponding to the keywords searched and rank assigned is given in the table (Table 6) (Fig. 4).

Table 1 Hardware configuration for proposed system

Node configuration	Description	Number of units
Server	Server hardware configuration	1
I3 Processor, 4 GB RAM, 500 GB HDD		
Server type	Local host	1
Client	Client hardware configuration	5
I3 processor, 4 GB RAM, 500 GB HDD		
System	Distributed	6

Table 3 Result obtained after user query pre-processing

Queries	User query	Tokenization	Dissimilar	Stemming	Spell check	Stop-words
Q1	In cricket a player uses a bat to hit the ball and scoring runs	In, Cricket, a, player, uses, a, bat, to, hit, the, ball, and, scoring, Runs	Cricket Uses scoring Runs	Score Use Run	Cricket	In, A, Player, Uses, A, To, The, and
Q2	I likes Apple	I, Likes, Apple	Likes	Like		I, Like
Q3	I am sits near the bank of river	I, am, sits, near, the, Bank, of, River	Sits Bank	Sit	Bank	I, am, sit, the, of

Table 4 Keyword extracted by the user query

Number of queries	Keywords	Sense annotation
Q1	Cricket score	–
Q2	Apple	Fruit, electronics
Q3	Bank	Bank, river

Table 5 A parametric result based on execution time and without clustering

No. of keywords fetched from user	No. of URL with KNN approach	Execution time with KNN approach (s)
Query		
Apple	180	1.03
Bank	300	1.85
Interest	160	1.08
Java	120	0.85
Orange	100	1.78
Giant	200	1.94

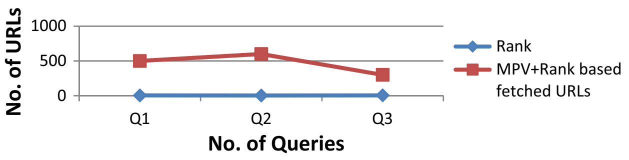

Fig. 4 Most probable clustering with rank allocation

As the rank is allotted and a cluster is formed hence the execution time required subsequently reduced. The number of keywords, although increased, but the execution time is reduced. Table 5 predicts the execution time with the clustering mechanism employed. Clustering setup locality of reference and allows a searching process to be simplified with the least complexity. Using the said mechanism only those clusters which are likely to contain the specified keyword is searched and the rest of the cluster are ignored causing least time consuming during the browsing of information (Table 7).

The execution time in browsing can cause mass users to attract the search engine or reject it. Execution time thus plays a critical role during browsing. The objective of the proposed browsing scheme is to reduce the complexity during searching for URL over the web. Google API's plays a critical role in our simulation work. Locations sensitive API's are employed in order to give the result specific to the location that also limits the search space causing reduced execution time.

Fig. 3 Keywords fetched without clustering from user query and execution time

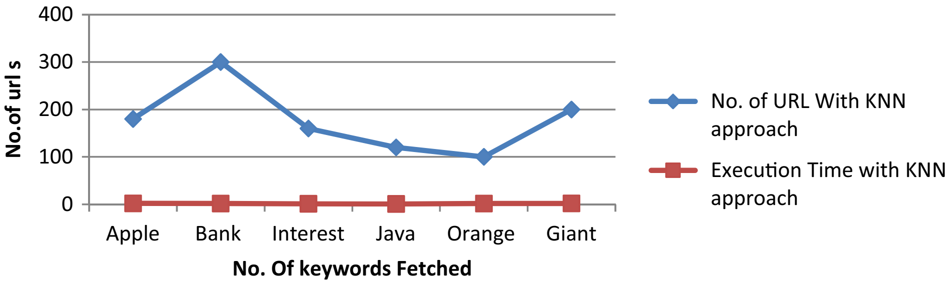

Table 6 Most probable clustering with rank allocation

Queries	Keywords	Clustering based fetched URLs	Rank based fetched URLs	MPV + rank based URLs
Q1	Cricket score	600	600	1
Q2	Apple	550	600	2
Q3	Bank	280	300	3

Table 7 Execution time with clustering

No. of keywords fetched from user query	No. of URL with clustering	Execution time with clustering (s)	TP (true positive) URLs	FP (false positive) URLs
Apple	700	0.85	300	400
Bank	800	1.03	400	400
Interest	120	0.65	100	20
Java	500	0.30	300	200
Orange	150	0.98	100	50
Giant	200	1.04	150	50

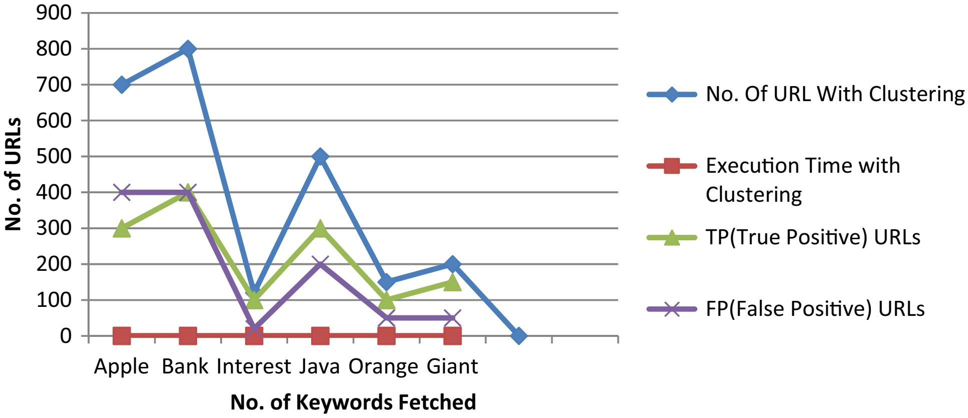

Fig. 5 Execution time with clustering

10.1 Discussion of result

The result obtained using the proposed system reduces execution time while fetching result from the server. Execution speed is enhanced by a significant margin. This is demonstrated using the simulation using ASP.NET. The keyword extraction phase and relevant URL fetched are shown in Fig. 5. The rank-based approach is compared against the hybridization of MPV and rank-based approach. A number of fetching keywords showing semantic analysis are limited, causing less execution time. Location sensitivity reduces the number of URLs fetched and hence reliability is enhanced.

11 Conclusion and future scope

In URL retrieval operation the volume of search space usage during web browsing results in high execution time with effect the reliability. Execution time reduction can cause mass users to interact with the browser. The aim of the proposed work is to reduce execution time by the use of the most probable clustering mechanism along with a user query correction mechanism makes it useful to look for meaningful and specific URLs. It also improves false positive and true positive rate. The result in terms of execution time, true positive rate and false positive rate with clustering shows improvement. Direct interaction of a user during word correction allows better communication and specific URL results. An additional advantage of the proposed mechanism is location-sensitive web URL is fetching that is obtained using the location API's provided by Google. In the future, the proposed work implication in a real-time environment can be tested and execution time can be further improved using a high degree of specificity through redundancy check and elimination procedure.

References

1. Sharma S, Sunita AK, Rana V (2018) An optimum approach for preprocessing of web user query. Int J Inform Commun Technol 7(1):8–12
2. Le QT, Pishva D (2015) Application of web scraping and Google API service to optimize convenience stores distribution. IEEE
3. Agre GH, Mahajan NV (2015) Keyword focused web crawler. In: IEEE International Conference on Electronics and Communication Systems
4. Rana V (2018) Optimizing performance of user web browsing search. In: International conference on advanced informatics for computing research, Springer, Singapore, pp 230–239
5. Zaman Z (2017) Spam detection in social media employing machine learning tool for text mining. IEEE Access

6. Witten IH, Moffat A, Bell TC (1999) Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann, Burlington
7. Sharma A (2014) Spam filtering using K mean clustering with local feature selection classifier. IJCA 108(10):35–39
8. Horecki K, Mazurkiewicz J (2015) For automatic prediction mechanism. Springer, vol 2, no 4
9. Rocha V, Kon F, Cobe R (2014) A hybrid cloud-P2P architecture for multimedia information retrieval on VoD services. Computing 98(1):73–92
10. Aguilar J, Valdiviezo-di P, Riofrio G (2017) A general framework for intelligent recommender systems. Appl Comput Inform 13:147–160
11. Pavalam SM, Raja SVK, Jawahar M, Akorli FK (2012) Web crawler in mobile systems. IJMLC 2(4):531
12. Naaz S (2015) Analysis of web pages through link structure. IJCA 122(11):22–26
13. Lee T, Chun J, Shim J, Lee S (2006) An ontology-based product recommender system for B2B marketplaces. Int J Electron Commer 11(2):125–155
14. Chen YS, Chang WH, Fang HM, Yeh YM, Cheng RS (2010) A context-aware reasoning framework with OWL for mobile web information acquisition. J Internet Technol 11(2):203–213
15. Cristina AH, Lopes CV, Givargis T, Atri M (2004) Location-aware web system. Work Build Softw Pervasive Comput Object Oriented Program Syst Lang Appl
16. Sharma S, Mahajan S, Rana V (2019) A semantic framework for ecommerce search engine optimization. Int J Inf Technol 11(1):31–36