Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130412

An Intelligently-Focused Crawling for Filtering the e-Learning Documents Using Optimized Hidden Naïve Bayes Classifier

A. Ramachandran[1,*] *and S. A. Sahaaya Arul Mary*[2]

[1] Department of Computer Science and Engineering, University College of Engineering, Panruti, India
[2] Department of Computer Science and Engineering, Saranathan College of Engineering, Trichy, India

Received: 13 Dec. 2018, Revised: 19 Mar. 2019, Accepted: 22 Mar. 2019
Published online: 1 Jul. 2019

Abstract: The major role of a Focused Crawler (FC) is to retrieve the related pages for the specific area and it avoids the unrelated links from the crawling queue. In this paper, we introduce the effective changes over the focused crawler by considering an area distiller. The domain distiller must check the links which are considered for embedding into the distiller queues. The proposed distiller is developed based on an Optimized Hidden Naïve Bayes (OHNB) classification algorithm which is the combination of the existing Hidden Naïve Bayes (HNB) and the Enhanced Multiclass Support Vector Machines (EMSVM). Here, the Genetic Algorithm (GA) is used for optimizing the soft margins of EMSVM. Then, this new optimized MSVM takes care to remove the outliers which are available in the training dataset. Experimental results prove that the efficiency of the proposed model when compared with other existing techniques.

Keywords: Enhanced Multiclass Support Vector Machine, Hidden Naïve Bayes, Optimized Hidden Naïve Bayes, Focused Crawler, Multiclass Support Vector Machine, Genetic Algorithm.

1 Introduction

Information Retrieval (IR) is a very challenging issue in this internet world due to the rapid development of internet users and the availability of huge volume of web applications for the particular context. Retrieving the specific content is a crucial task today from this large volume of information which is retrieved from the respective context information [1]. In this scenario, the Search Engine (SE) services are very important today for extracting the specific content in any specific direction. Moreover, the efficiency of the search engine plays major role in this internet era. Even though, the use of web crawler which is the main component of the SE is very important for identifying the suitable web pages. The related information is passed by the web crawler to the indexer for easy retrieval.

Focused Crawler (FC) is something special than the conventional web crawler. Because the FCs is used to download the relevant pages only which are more suitable for a specific DOI [2]. In this process, the web page is divided into a set of relevant and interrelated areas and also more than one crawler is permitted to identify the specific domain. Moreover, web page indexing process is done parallel according to the web content coverage. Even though, it has number of challenges and unsolved problems. From these issues, the Haste Problem issues are considered as effective and major issues which focus on the efficiency of crawling. This problem occurs when the bad links are injected into the crawling queue of the FC. These more bad links are added to the queue which causes the FC to be skewed away from the respective DOI. Here, the FC is used to estimate the similarity of the DOI.

Even though, the FCs use many accurate estimation methods in the past for estimating the relevancy. But, their relevancy prediction is not accurate most of the times. For overcoming this kind of challenging issue, the various mining algorithms have been proposed by many researchers for calculating the exact relevancy score according to the presence of actual web contents which are available in number of actual web pages. For

* Corresponding author e-mail: hariprasadsec@aol.com

achieving the aim of FC in this direction, a domain distiller is deployed for computing the relevancy score of the specific web page after retrieving it. Then the decision to pass the retrieved page to the search engine's indexer or to add the page's embedded links to the crawling queue is based on the distiller's decision [3].

In e-Learning, the web content identification and classification are very important for retrieving the relevant content for the specific topic of a subject. Generally, classification can be classified into two: two classes and multi classes. Here, two different sets of e-documents can be considered as training and testing. Such e-documents are used to train by using the classification rules which are framed by the administrator of the system. Here, the particular e-content-related domain distillers can be designed as two-class classification which is related to a specific DOI. Many classification techniques have been introduced for performing two-class and multiclass classification. For solving two-class problem, many two-class classifiers such as Support Vector Machines (SVM) [4], decision trees [6], K-nearest Neighbor (KNN) [5], neural networks [7] trees [6], K-nearest Neighbor (KNN) [5], neural networks [7] and Naïve Bayes [8] are used. Similarly for solving multiclass problems, one uses multiclass classifiers such as MSVM, EMSVM, IAEMSVM and IREMSVM.

The authors proposed a new focused crawler named OntoCrawler which provides an ontology supported and semantic-based solution. Hence, their crawler can benefit both the user requests and the domain semantics. Their focused crawler has been tested with Yahoo and Google search engines to search the related information [8].

A natural language processing method to content-based video indexing and retrieval for identifying the suitable video clips which is capable of addressing the user's needs. Their approach integrates the natural language processing, the entity extraction method, frame-based indexing technique, and the information retrieval methods to extract the knowledge in an e-learning environment [9]. Reusability tree for representing the relationships among the relevant objects and the enhanced CORDRA. They have collected the relevant data and the learning objects like citations and time period. Moreover, they introduced a new mechanism for assigning the weights and rank to the learning objects. They also provided a tool named 'Search Guider' for assisting the users in searching the relevant data in citations and time period-based on the individual users requirements [10].

A new information retrieval model is proposed for rectifying the flaws of the ranking algorithms and also for improving the capability of web search engines. Today, the search engines do not rank the documents which are searched for the specific query and also it simply retrieved the related documents. In their model, the user can efficiently access previous history results with minimum clicks and access recently preferred results in ranked order [11]. Design and the functionality of the *R* Crawler which is the combination of *R* language and Crawler.

A new re-ranking method recognizes the most relevant web information to feed in the specific and relevant area knowledge learning hub. Moreover, their method studied the structure and semantics of the ontology and also designs the computational relationship among the nodes. They have calculated the three dimensional data scores such as distance, direction and the features of each document and the re-ranked the documents which are retrieved from web that provides the learners with meaningful knowledge in the respective area [13]. The literature study proposes a recommendation system which is developed using the ontology and the dimensionality reduction techniques for improving the scalability. They also discuss the prediction accuracy and the time complexity of the proposed model [14].

The strategies of the information retrieval in web crawling has been presented which is categorized into four methodologies such as focused web crawler, distributed web crawler, incremental web crawler and the hidden web crawlers. Finally, the comparative analysis between the various IR strategies and the existing IR models strategies [15]. The performance of supporting key and quality tasks can be evaluated by computing parameters like eliminate the data quality levels, prediction about decline the quality, cost adjustment and the benefits which are associating with the correcting flawed data, optimize the data audit and the requisition policies. Their experimental results highlight the major contributions such as enhancing the cost effectiveness of the data acquisition process, policy management and reducing the damage which is related to the flawed data [16, 17].

In this paper, we have proposed a new method for channel distiller using Optimized Hidden Naïve Bayes (OHNB) classification algorithm which is the combination of the existing Hidden Naïve Bayes (HNB) and the Enhanced Multiclass. This paper is organized as follows: Section 2 provides the detailed description of the proposed channel distiller. Section 3 provides the training phase Then the performance of the proposed technique is evaluated in Section 4 and Section 5. Finally Section 6 concludes the paper.

2 Proposed Work

2.1 Proposed Domain Distiller

The proposed domain distiller is a web page to be classified. It analyses web pages and extracts the existing related domain key terms using the existing Disambiguation Domain Ontology (D2O) that maintains all the available related domain key terms. Even though, before mapping the related and extracted domain key

terms to the respective concepts. Here, the disambiguation process is also done over the extracted related domain key terms which are to discard those ambiguous keywords that are not related to the related domain. The main aim of the proposed domain distiller is to decide whether the input web page if it is according to the domain interest or not. Since the proposed domain distiller acts as a binary classifier which has two classes. Hence, the proposed domain distiller is also divided into two modules such as analysis module and the classification module. In analysis module, the input web page is explained in detail about the vector space modelling. In classification module, it decides to accept the web page or reject it using the proposed OHNB classifier.

The main aim of analysis module is to represent the input web page in the vector space modelling. For that purpose, the related domain's key terms are identified in the web page and these key terms are extracted and also mapped with the respective domain concepts using the disambiguation domain ontology. Earlier to this, based on the domain of interest k key terms are clustered by using the WordNet. Moreover, the keywords are grouped based on the synonyms matching and represented by a specific domain concept.

The main objective of the classification module is to classify the web pages in two-class fashion as a binary classifier. The web pages may be related to the domain of interest or it may not. Here, a new binary classifier is introduced by integrating an optimized Multiclass support vector machine with the use of existing GA and the HNB algorithms. However, the optimization process has been done on MSVM, such process also optimizes the behaviour of HNB. Hence, the new optimized binary classifier is the core part of the proposed domain distiller is named OHNB classifier. Generally, it rejects the outliers by selecting the most useful examples which use an optimized MSVM with GA. Hence, after the rejection of outliers, the remaining web pages are to be used and to train the HNB classifier. Finally, the OHNB takes care of the web page classification as "Class1" or "Class 2" for testing. The OHNB works in three phases such as outlier rejection, training and testing. These three phases are explained in the proposed OHNB algorithm.

Outlier rejection is used in this work to discard the false samples which may result in constructing the wrong classification during training. Generating wrong rules affects the performance of the system badly and it turns the outlier rejection is used in this work to discard the false samples which may result in constructing the wrong classification during training. Generating wrong rules affects the performance of the system badly and it turns the focused crawler in wrong direction.

Outlier rejection is performed through Optimizing SVM and by selecting the useful web pages. Here, the genetic algorithm is an important, effective, powerful and unbiased heuristic search method in artificial intelligence. First, it has many advantages such as the ability to solve any optimization problem using chromosome technique. Second, it handles many solutions perfectly. Third, it is less difficult and more straightforward when compared to the classifiers. Fourth, it is easy to transfer and it is applied in various platforms-based on the system flexibility level. Fifth, it is also able to perform multi-objective optimization. Moreover, it has the capability to calculate the global optimum.

The accuracy of testing dataset using the formula in Eq. (1).

$$FT_i = ACC_i = \frac{\text{Right Documents}}{\text{Total Web Pages}} = \frac{A_i + CR_i}{A_i + B_i + CR_i + ICR_i} \tag{1}$$

Here, FT_i represents the fitness value of the ith chromosome and the accuracy for the same chromosome is indicates by ACC_i, the number of pages which are assigned correctly using the same chromosome id is represented by A_i and the number of documents is rejected for the same chromosome expressed by CR_i and ICR_i which indicates the number of documents incorrectly rejected. Hence, the probability of each chromosome is selected is calculated by using the formula which is given in Eq. (2).

$$PR_i = \frac{FT_i}{\sum_{i=1}^{x} FT_i} \tag{2}$$

where P_i indicates the selection probability of the chromosome i, x is the number of all chromosomes in the population, and FT_i is the fitness value of chromosome i. Finally, the sum of all probabilities of all chromosomes is calculated using the formula which is given in Eq. (3).

$$TotPR_i = \sum_{i=1}^{x_n} PR_i \tag{3}$$

where, $TotPR_i$ indicates the sum of all probabilities of all chromosomes, x_n represents the number of all chromosomes which are available in the population, and PR_i indicates the probability value of the given ith chromosome. Then, a new Random Number (RN_n) from the interval $(0, TotPR)$ is generated as shown in Eq. (4).

$$RN_n = RNand(0, TotPR) \tag{4}$$

where, RN_n is a random number between the 0 and $TotPR$, and $TotPR$ indicates the total probability values of all chromosomes which are available in the population. Then, it has considered the population again to accumulatively as total probability value of from 0 to $TotPR$. During total probability value calculation for all chromosomes, if $TotPR$ reaches a value which is greater than or equal to the random number RN_n which is generated by using Eq. (4) then, the process must be stopped and select the particular chromosome.

3 Training Phase

In this training phase, traditional Naïve Bayes classification algorithm has been trained by using the most informative sample documents and examples which are selected during the process of the previous phase. Therefore, a traditional Naïve Bayes classification algorithm has been deployed having two-target classes represented by the set $CLS = \{cls_1, cls_2\}$, where cls_1 is the class which which also includes the input web pages that are related to the domain of interest during cls_2 is to be the opposite class in this process. During Naïve Bayes classification algorithm training, the task is to calculate the conditional probability values $P(cs_i \mid cls_j) \forall cs_i DC$, $cls_j DCLS$, and also the classes which have the prior probability values $P(cls_j) \forall cls_j DCLS$ as shown in Eq. (5) and Eq. (6) respectively.

$$PR(cls_j) = \frac{PRg_j}{PR_g} \tag{5}$$

$$PR(cs_i \mid cls_j) = \frac{N_{i,j}}{N_j} \tag{6}$$

where, PRg_j indicates the number of web pages that are related to the class cls_j, and the PR_g represents the total number of input web pages that are related to all the domain classes, $N_{i,j}$ indicates the total number of occurrences of the target concept in the various pages of a class cls_j, and N_j represents the total number of concepts available in the input web pages of a class cls_j.

3.1 Testing Phase

In this testing phase, the decision can be taken about the input web pages whether an input page is related to the domain of interest or not. If the input web page is related then, the input web page is targeted to the cls_1 in this process. Otherwise, cls_2 is the targeted web page and it must be rejected in this process. Initially the probability value pri that is related to each class is calculated, it is also called the likelihood of being in the class according to the domain concepts that are extracted from pr_i. If pr_i contains zc domain concepts and the likelihood of being in class $cls_j \forall cls_j \in CLS$ is calculated and shown in Eq. (7). Moreover, pr_i is targeted over the $clstarget$ class with the maximum calculated value of likelihood (LH) by using Eq. (8).

$$LH(PR \mid cls_j) = PR(cls_j) + \prod_{x=1}^{z} PR(cs_{x,i} cls_j) \tag{7}$$

$$\begin{aligned} Target(pr_i) &= cls_{target} \\ &= \arg\max(PR(cls_j) + \prod_{x=1}^{z} PR(cs_{x,i} cls_j)) \end{aligned} \tag{8}$$

Algorithm: Optimized Hidden Naïve Bayes Classifier

Inputs $CLS = \{cls_1, cls_2, \ldots, cls_m\}$ // Predefined m number of classes
$TSs = \{ts_1, ts_2, \ldots, ts_i\}$ // Training sample documents
Output A class that has the number of web pages.

Step 1: Let assume the various chromosomes that are available and considered.
Step 2: Find the fitness value for every chromosomes which are considered in this work using

$$FT_i = ACC_i = \frac{\text{Right Documents}}{\text{Total Web Pages}} = \frac{A_i + CR_i}{A_i + B_i + CR_i + ICR_i}$$

Step 3: Indicate each and every chromosome in binary value.
Step 4: Compute the probability value for selecting each chromosome using the formula

$$PR_i = \frac{Ft_i}{\sum_{i=1}^{i} Ft_i}$$

Step 5: Find the total probability value of all chromosomes using the formula

$$TotPR_i = \sum_{i=1}^{n} PR_j$$

Step 6: For each chromosome I in population Do
Step 7: Generate random number (RNn) in the range of $[0, TotPR]$ using the formula

$$RNn = Random(0, TotPR)$$

Step 8: The total probability value is calculated by using the formula

$$TotPR = TotPR + PR_i$$

Step 9: If $(TotPR >= RN_n)$ Then
Stop and select chromosome i.
Step 10: End If
Step 11: End For
Step 12: For each chromosome I in population Do
Step 13: Generate random number (RN_n) from (0 to $TotPR$) using the (4).
Step 14: If $(RN_n < PR_c)$ Then
Stop and select the chromosome i as a parent
Step 15: End If
Step 16: End For
Step 17: Select the crossover point within a chromosome randomly.
Step 18: Interchange the two parent chromosomes at this point to produce the offspring.
Step 19: Call the EMSVM (Ganapathy et al. 2012) to train the selected chromosome.

Step 20: Finalize the margin in EMSVM for making decision over the chromosomes.
Step 21: Remove the chromosomes which are available in out of the margin.
Step 22: Call Hidden Naïve Bayes (HNB) classification algorithm for training with the informativeweb pages that are extracted from dataset.
Step 23: Compute the conditional probability values for all chromosomes using the formulas:

$$PR(cs_i \mid cls_i) \forall cs_i \in CS, cls_j \in CLS$$

$$PR(cs_i \mid cls_i) = N_{i,j}/N_j$$

Step 24: Compute prior probability values for the classes using the formula

$$PR(cls_j) = PRg_i \in PRg$$

Step 25: Compute the probability value for the web pages pr_i which are related to all the classes (LH) one by one using the formula:

$$LH(pr_i \mid cls_j) = PR(cls_j) \times \prod_{x=1}^{k} PR(cs_{ij} cls_j)$$

Step 26: Choose the target class for the particular web document page $pri(cls_{target})$ with the maximum calculated value of LH using the formula:

$$Target(pr_i) = cls_{target} = \arg\max(PR(cls_j) \times \prod_{i=1}^{k} PR(cs_{ij} cls_j))$$

Step 27: Return a class which have number of web pages.

The proposed optimized hidden Naïve Bayes classification algorithm has three different phases such as optimization phase is optimized by using the genetic algorithm, training phase which is responsible for train the web pages using the exiting classification algorithm called EMSVM and testing phase which uses the probability value for the web pages for selecting the valuable web pages. Finally, it returns the valuable web pages which are useful for understanding the e-contents for the e-learners to learn easily. This optimized classifier has been built with the domain distiller which is very useful in the focused crawler for extracting the relevant web pages.

4 Results and Discussion

In this section, the proposed domain distiller has been evaluated against the standard classification algorithms such as SVM, Naïve Bayes, K-NN and some domain-oriented Naïve Bayes classifier and Domain-Oriented K-NN classifier. Here, we have considered three evaluation metrics such as accuracy, precision and error. The web data common dataset series contains all structured data extracted from the various common crawl corpora. Here, the data formats like Microdata, RDFa and microformats have been considered for extracting the relevant information. Moreover, the web documents which can be used as e-learning documents of web data commons are not pre-designated as training patterns or else testing patterns. For evaluating the proposed model, we have considered 10000 web pages that have been related to e-learning documents for a subject as training and testing subsets. From these 10000 web pages, we have selected randomly only 500 web pages for training and 500 web pages are used for testing.

4.1 Performance Metrics

The proposed domain distiller model has been evaluated based on the three performance metrics such as precision, accuracy and error using Eqs. (9), (10) and (11) respectively.

$$\text{Precision} = P = \frac{\text{Pages assigned correctly}}{\text{Total assigned pages}} = \frac{A}{A+B} \quad (9)$$

$$\text{Accuracy} = Acc = \frac{\text{Corrected assignments}}{\text{Total pages}} = \frac{A+C}{A+B+C+D} \quad (10)$$

$$\text{Error} = E = \frac{\text{Incorrect assignments}}{\text{Total pages}} = \frac{B+D}{A+B+C+D} \quad (11)$$

5 Experimental Results

In this section, we have discussed about the performance of the proposed classifier on the newly-proposed domain distiller through various experiments with different number of web pages which are extracted as input page for e-learning in this work. Here, we have considered the precision, recall and error have been calculated by using Eqs. (9), (10) and (11). First, the precision value is calculated for the different set of web pages such as 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 and 10000.

Fig. 1 shows the performance of the proposed model and the existing classification algorithms such as K-NN, DOK-NN, SVM, NB, ONB, DONB, HNB and OHNB. Here, we have conducted ten experiments with various numbers of web pages which are useful for learning a subject through e-learning and also to extract the relevant content from the internet.

From Fig. 1, it can be observed that the precision value of the proposed OHNB model is high when it is compared with the existing classification algorithms such as K-NN, DOK-NN, SVM, NB, ONB, DONB and HNB. This is due to the use of hidden Naïve Bayes classifier.

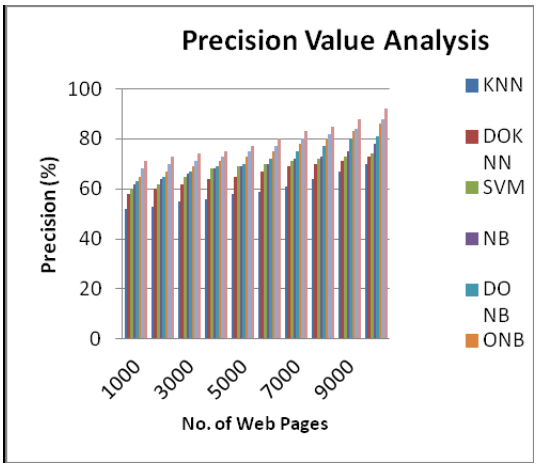

Fig. 1: Precision value analysis

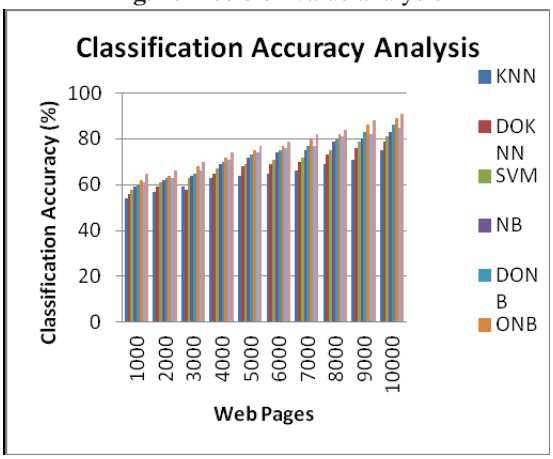

Fig. 2: Classification accuracy analysis

Fig. 2 shows the classification accuracy of the proposed model and the existing classification algorithms like K-NN, DOK-NN, SVM, NB, ONB, HNB and OHNB which are proposed in this direction in the past by various researchers. Here, we have conducted ten experiments with different number of records between 1000 and 10000.

From Fig. 2, it can be seen that the classification accuracy of the proposed OHNB model is performed well when it is compared with the existing classification algorithms such as K-NN, DOK-NN, SVM, NB, ONB and HNB. This is due to the use of effective fitness function on GA and the use of Hidden Naïve Bayes Classification algorithm in the optimized form.

Fig. 3 shows the error analysis for the proposed model and the existing classification algorithms such as K-NN, DOK-NN, SVM, NB, ONB and HNB. Here, we have considered 1000 to 10000 retrieved web pages with equal number interval number of web pages for evaluating the proposed and exiting system.

From Fig. 3, it can be observed that the error rate of the proposed model is very low when it is compared with the existing systems such as K-NN, DOK-NN, SVM, NB,

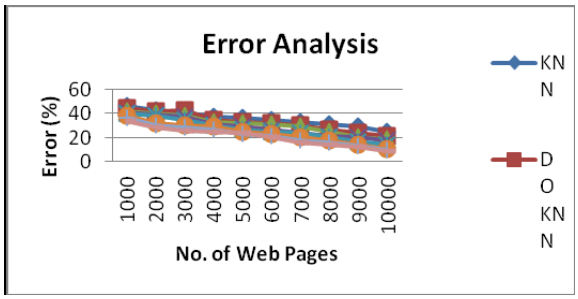

Fig. 3: Error analysis

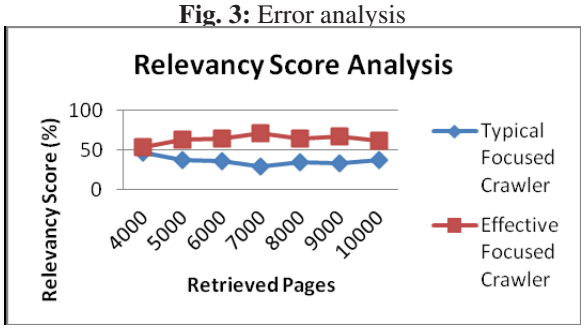

Fig. 4: Relevancy score analysis

ONB and HNB. This is due to the fact that the uses of effective fitness function over the optimization, the Enhanced SVM classifier and the proposed Optimized Hidden Naïve Bayes classification algorithms.

Fig. 4 shows the performance of the relevancy score for the proposed focused crawler and the standard focused crawler. Here, we have considered the various number of retrieved web pages such as 4000, 5000, 6000, 7000, 8000, 9000 and 10000. The relevancy score is calculated based on the standard formula in this work.

From Fig. 4, it can be observed that the proposed focused crawler is performed well than the existing standard focused web crawler. This is due to the use of effective classification by using the existing classifiers such as enhanced multiclass SVM and the hidden naïve Bayes classification algorithms. These classification algorithms are used to improve the performance of the proposed model.

6 Conclusion

In this work, a new and effective area domain distiller has been developed by using the proposed Optimized Hidden Naïve Bayes (OHNB) classification algorithm for selecting the most informative web pages for learning a subject through e-learning. This new domain distiller has been adopted with the focused crawler for retrieving the useful web contents from the internet source or datasets. This focused web crawler has been developed with the slight modification over the existing typical web crawler. Here, new characteristics have been added for improving the performance of the domain distiller.

The proposed focused web crawler extracts the relevant information effectively by using the proposed classification model called Optimized Hidden Naïve Bayes (OHNB). Here, we have used the GA for optimization and EMSVM for training the web pages effectively. The experimental results have proved that the effectiveness of the focused web crawler in terms of accuracy which indicates the number of documents correctly identified and extracted by the crawler and the relevancy score which is calculated based on the relevant web page documents.

References

[1] I. Ahmed, E. Saleh, Arwa, F. Abulwafa, Mohammed and Al Rahmawy, A web page distillation strategy for efficient focused crawling-based on optimized Naïve Bayes (ONB) classifier, *Applied Soft Computing*, **53**(5), 181–204 (2017).

[2] A. Elyasir and K Anbananthen, *Focused web crawler*, International Conference on Information and Knowledge Management, **45**, 149–153 (2012).

[3] Liangxiao Jiang, Harry Zhang and Zhihua Cai, A Novel Bayes Model: Hidden Naive Bayes, *IEEE Transactions on Knowledge and Data Engineering*, **21**(10), 1361–1371 (2009).

[4] A. Sun, E. Lim and W. Ng, *Web classification using support vector machine*, Proceedings of the 4th International Workshop on Web Information and Data Management, New York, ACM Press, 96–99 (2002).

[5] S. Ganapathy, P, Yogesh and A. Kannan, Intelligent agent-based intrusion detection system using enhanced multiclass SVM, *Computational Intelligence and Neuroscience*, **2012**(1–10), 1–10 (2012).

[6] R. Navigli, Word sense disambiguation: a survey, *ACM Computing Survey*, **41**(2), 10.1–10.69 (2009).

[7] M. Jamali, H. Sayyadi, B. Hariri and H Abolhassani, A method for focused crawlingusing combination of link structure and content similarity, *Web Intelligence, IEEE Computer Society*, 753–756 (2006).

[8] S. Yang, OntoCrawler: a focused crawler with ontology-supported website models for information agents, *Expert Systems and Applications*, **37**(7), 5381–5389, (2010).

[9] Dongsong Zhang and Jay F. Nunamaker, A Natural Language Approach to Content- BasedVideo Indexing and Retrieval for Interactive E-Learning, *IEEE Transactions on Multimedia*, **6**(3), 450-458 (2004).

[10] Neil Y. Yen, Timothy K. Shih, Louis R. Chao and Qun Jin, Ranking Metrics and Search Guidance for Learning Object Repository, *IEEE Transactions on Learning Technologies*, **3**(3), 250-264 (2010).

[11] S. Foram, Manek, J. Aishwarya Reddy, Vaibhavi Panchal and Vijaya Pinjarkar, *Hybrid Crawling for Time-Based Personalized WebSearch Ranking*, International Conference on Electronics, Communication and Aerospace Technology, 252–255 (2017).

[12] Salim Khalil and Mohamed Fakir, RCrawler: An R package for parallel web crawling and scraping, *SoftwareX*, **6**, 98–106 (2017).

[13] Grace Zhao and Xiaowen Zhang, *A Domain-Specific Web Document Re-Ranking Algorithm*, 2017 6th IIAI International Congress on Advanced Applied Informatics, 385–390 (2017).

[14] Mehrbakhsh Nilashi, Othman Ibrahim and Karamollah Bagherifard, A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques, *Expert Systems With Applications*, **92**(2), 507–520 (2018).

[15] ChandniSaini and VinayArora, Information Retrieval in Web Crawling: A Survey, Intl. Conference on Advances in Computing, *Communications and Informatics (ICACCI)*, 2635–2643 (2016).

[16] Yuval Zak and Adir Even, Development and evaluation of a continuous-time Markov chain model for detecting and handling data currency declines, *Decision Support Systems*, **103**(11), 82–93 (2017).

[17] WalidMaalej, Mathias Ellmann and RomainRobbes, Using contexts similarity to predict relationships between tasks, *The Journal of Systems and Software*, **128**(6), 267–284 (2017).

A. Ramachandran is working as Assistant Professor in the Department of Computer Science and Engineering, University College of Engineering Panruti, Tamilnadu, India. He completed his Master of Engineering in Annamalai University and pursuing Ph.D. in Anna University. His areas of interests are Software Engineering, Data Mining and Web Crawlers.

S. A. Sahaaya Arul Mary is working as Professor and Head in the Department of Computer Science and Engineering, Saranathan College of Engineering, Trichy, Tamilnadu, India. She has completed Ph.D. in Anna University. She has more than 25 years of teaching and research experience. Her areas of expertise are Software Testing, Artificial Intelligence and Data Mining.