IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Preprocessing and Classifying Web Text Data for E-learning Recommendation

Kamika Chaudhary[1*], Neena Gupta[2]

[1] Department of Computer Science, Kanya Gurukul Campus (Gurukul Kangri Vishwavidyalaya) Dehradun, India
[2] Department of Computer Science, Kanya Gurukul Campus (Gurukul Kangri Vishwavidyalaya) Dehradun, India

[*]*Corresponding Author: kamika.agrohi@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 22/Jun/2018, Published: 30/Jun/2018

Abstract— Growing competition over the years has seen an increase in getting vital information like customer behaviour, his likes and dislikes before launching a product. Extracting the information from a huge pool of data like internet is what we in technical terms know as Web Mining (WM). With the technology comes the challenges too and getting correct information from a very large pool of data is always a big task. Traditionally WM uses content, structure and usage mining techniques but still the user sometime is not able to retrieve what he is looking for. Proper filtering of the information retrieved in the form of text or in other words text mining could make a lot of difference between correct information and lot of information. The paper focuses on digging the web to create a comprehensive repository for web miners looking for e-learning. 2000 URLs related with different online learning were taken into consideration, the information was read using python and raw text was collected. Python's punctuation and itemgetter modules were used to retain only the major keywords having counts over a threshold, after performing basic text mining techniques. To check the robustness of the retained data precision, recall and accuracy was calculated and it was found that the precision, recall and accuracy were 0.964, 0.982 and 0.97 respectively.

Keywords—Web Mining, Text mining, E-learning

I. INTRODUCTION

The rapid advancement in information and communication domain and technologies in recent years has caused the increase in usage of E-learning systems. Due to high utility of e-learning systems, a huge volume of learning resources also gets proliferated in an explosive manner. This gigantic accumulation of information in the form of online learning resources presents a difficulty to user in selecting relevant learning resource. E-learning systems comes face to face with the issue of information overload so it becomes difficult for learner to reaches to their most suitable and needed content. To deal with this problem recommender system plays the role of savior by retrieving the relevant learning material on the basis of personalized learner preference. These system works by filtering and recommending the appropriate learning resources to learners. They mainly work towards guiding the learner towards their interesting learning material and also focus on presenting right material to the intended learner at the correct time irrespective of the place.

Generally two recommendation techniques collaborative and content based filtering are utilized by e-learning recommendation system to provide effective recommendations [1]. Content based approach consider items or contents as their focal point and examine the properties of these items in predicting the next useful item whereas collaborative filtering finds the similarity among users and their accessing patterns and they recommend those items which are preferred by similar users. These traditional approaches suffer with cold start problem as well as sparsity and scalability issues. These drawbacks have caused the emergence of newly developed techniques for recommending learning resource to learner by exactly capturing and considering learner requirement of learners in e-learning systems. As each learner have different requirements depending on their learning style, knowledge level or learning goals, an open environment is required in which he can express his specification in natural language without any constraints [2].

It has been found that a large number of techniques have been used previously ranging from data mining, web mining to artificial intelligence and machine learning algorithms.

Here in this paper Knowledge Discovery in Text (KDT) or text mining domain is explored and used in order to propose a recommendation system for e- learning. Text mining is a sub area of web mining which makes use of specific techniques in order to uncover hidden but potentially important patterns which lies in unstructured textual data in logical units of text or documents [3]. Most of the mining techniques are based on collaborative filtering, having two phases neighborhood formation and recommendation phase but text mining based recommendation system does not consider neighborhood formation, it works on text input of the users rather than some hypothetical assumption. So in this work a text based approach for mining the web and

finally creating a classified e-learning repository is in the focus of the current paper. Since standard classifiers' are based on collaborative filtering a more specific and robust technique is required to suggest correct learning sites to the surfers.

The organization of paper follows the further mentioned paper, Section I describes the introduction of text mining and information overload problem in E-learning recommender system considered in this paper, Section II contain the related work of e-learning recommender systems, Section III contain the proposed work along with mathematical model and algorithm Section IV contains the performance analysis of proposed algorithm and Section V concludes research work with future directions

II. RELATED WORK

Nowadays with the increase in use of internet by everyone there is a huge competition that arises among business organizations in terms of making their web services more useful for users. One way of achieving consistent interest of user in the web enabled service is by presenting the most required information in front of him so that he does not wanders for his information. Another way is providing recommendations to him on the basis of his interests. It has also become an important requirement to provide personalized knowledge in the area of E-learning as the number of resources providing online learning services are present in a huge number. So in order to fulfil student's requirements it is important to figure out user's interests precisely and provides real time recommendations. A large variety of techniques have been used to provide recommendation in E-learning systems, one of them is presented in [4] by authors. They have used text mining based topic model in order to build user interest model using which recommendations are provided to users. Major focus is paid on interests modeling and an interest based algorithm had been proposed and its experimental evaluation had also been done and proposed method is found to be highly effective and adaptive.

In Jun et al. [5] authors have focuses on the need of learners in finding their right learning materials quickly so they presented a personalized recommendation system for e-leaning in online courses. An intelligent learning system consisting of three modules which are data support module, combinational algorithm based recommendation engine module and new source recommendation module had been proposed. The combination algorithm module comprises association rules, content filtering and collaborative filtering algorithm. The results after performing experimental evaluation of this system indicates towards improvisation in utilization rate of learning resources as well as efficiency of students.

Mostly traditional recommender system works by utilizing content based filtering and collaborative filtering techniques. They do not focus on the context while making recommendations so resulted into making inappropriate recommendations and also suffers from data sparsity problem. Learner's context information plays a vital role as learner preferences shifts from one context to another context so authors in [6] presented a hybrid approach for e-learning recommendation system depending on context awareness and sequential access patterns. Main aim behind using context awareness is to include learner's contextual information like level of knowledge and goal of learning while sequential pattern mining algorithm had been used to uncover the sequential access patterns of learners and finally at last both these approaches are added to collaborative filtering in order to provide most relevant recommendations. These algorithms are applied on the data set collected from a university learning management system and experimental evaluation and comparison with proposed approach shows that the purposed method outperforms existing methods.

The problem faced by learners in searching the learning material matches to their requirements is due to the availability of resources in huge amount. So to tackle this information overload issue a new method depending on knowledge based reasoning and collaborative filtering is given in [7]. In this a unified system named as weighted hybrid system of rule-case based reasoning and matrix factorization benefitting both teacher and student is presented. The experiments is performed on three datasets taken from an intelligent tutoring system called as cognitive tutor and results proves to show better prediction accuracy.

It has been seen that in most of the recommender system courses and materials are static in nature and dynamic aspect exist only in the organization of the material. Authors in [8] proposed an adaptive learning system which is dynamic in nature for both learners and open web by making use of data mining techniques such as data clustering. The system consists of paper maintenance module in which a web crawler is used to find the course related paper and updated the paper repository. Second module is for recommendation which includes data clustering module and collaborative filtering module. Experimental evaluation depicts degradation in computational cost without hampering the overall performance of the system.

Another recommendation system for recommending books by utilizing data mining algorithm such as Bayesian algorithm has been presented in [9]. In this text categorization techniques are applied on the semi structured text which is collected from the web. Mainly content based recommendation is applied in order to provide book titles as suggestions on the basis of individual user training data. This approach also works efficiently even in the absence of any information about other users.

A framework for e-learning system which focuses on increasing the student efficiency in learning environment by incorporating their learning style is discussed in [10]. Learning styles incorporation plays an important role as every student has different learning goals as well as behavior so recommendation should be provided by considering these

factors. In the proposed system a user modeling structure which is built by using latent semantic analysis technique is present. Then a rule based expert system was used to provide recommendations and it resulted in improving student efficiency as well as productivity.

Another hybrid recommendation system which includes multi model ontology was proposed by authors in [11]. Domain ontology and rule based ontology are incorporated in order to provide semantic recommendation for e-learning system. Top-n-recall and Top-n-precision had been used as metrics for finding the effectiveness of experimental evaluation that has been performed. An improvement of 5-20% has been noticed that proved the effectiveness of the system. The concept of semantics is further studied and used in e-learning recommendation system in [12]. A framework for recommendation system is proposed by utilizing semantic algorithm which is based on intra and extra semantic relationship existing among learning objects and learners needs.

Further the issue of sparsity in collaborative filtering based recommendation system was dealt by introducing content based filtering along with it in e-learning recommendation system [13]. In this analysis of web content is performed in order to compute content based document similarity of information items. This system helped student in sharing their knowledge and interests and their learning process is not limited to only classroom study sessions.

Further the process of web mining particularly web usage mining has been used in proposing a framework that provides automatic recommendation without asking for explicit feedback from user [14].The system composed of two separate modules that is offline and online module . Clustering, association rule mining and inverted index matching techniques are employed in recommendation phase of the systems. Precision and recall have been used as evaluation metrics which plays their role perfectly and helped in providing the efficiency of the system.

An item based recommendation system for e-learning by utilizing ontology and genetic algorithm has been presented in [15]. Genetic algorithm plays an important role in optimizing the recommendation results. The system helped in solving the cold start problem and experiment results outperforms previous versions. Another approach that also makes use of genetic algorithm for group of learners is discussed in [16]. It has been found that sometimes the decision of recommendation can be influenced by a group of user making use of that system which resulted in to better prediction results. A profile merging scheme has been proposed by utilizing genetic algorithm in order to construct a unified learner model and then collaborative filtering is applied based on learning style and knowledge levels. The result shows improvement in performance.

III. PROPOSED WORK

To figure out the problem related with search engines not able to return the desired results a code in python was written to retain the meta description of the web sites. It was observed that for many sites only titles were returned and in some sites the keywords set by the web designers were returned which were actually of very little importance for a web miner looking for e- learning source. Figure 1 shows an output of one such query

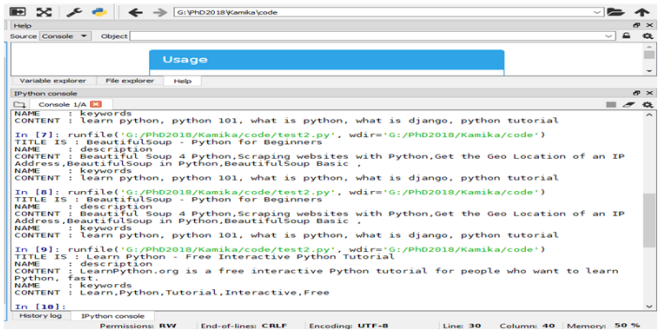

Figure 1. Meta description of a website retrieved from a url

To overcome the problem of search results an e-learning based classified repository is required. For the proposed work 2000 sites having contents for online learning were taken into consideration. A new classification algorithm has been designed on the proposed mathematical model.

Mathematical Model

In the proposed model $Kw_{collect}$ represents the list of keywords which are most commonly used related to several topics from e-learning of computer science subjects such as java, C#, python, jython, Go etc.

$$Kw_{collect} = \bigcup\{java, C\#, Python, Zython, Go, ruby, SE\ldots\} \quad \ldots\ldots(eq1)$$

$$URL_{content} \Leftarrow text \in URL_{address} \quad \ldots\ldots eq(2)$$

$$URL_{tokenized} = \{t \mid t \in URL_{content}, t \text{ is } text\} \quad \ldots\ldots eq(3)$$

Similarly $URL_{content}$ and $URL_{tokenized}$ are computed by using equation2 and 3 respectively where $URL_{content}$ represents all those urls in which the related contents are available and then those content is tokenized $URL_{tokenized}$ in which t represents the extracted text element of the URL contents.

$$\{Kw_{url}\} \Leftarrow \sum_{i=0}^{l} URL_{tokenize} if \in Kw_{collected} \quad \text{..... eq(4)}$$

Then by using equation 4 tokenized contents of the URL are looked into the classified collection of keywords

$$\{URL_{ij}, kw\} \Leftarrow \bigcup_{i=1}^{n} Kw_{url} \quad \text{..... eq(5)}$$

Finally equation 5 is the collection of URL and keywords retained. Here, n is the number of URLs visited.

Algorithm

Step 1: $Kw_{collect}$ ← Collect keywords with maximum hit counts of different topics on e-learning

Step2: Loop till true

url_content ← read URL related with a topic, example Python.
F_content ← using text mining techniques –
a. Basic cleaning
b. Parse text
c. Remove special characters
d. Perform lexical analysis

C_File ← Create clean text file.
Split_content ← Read c_file and split its contents
Token_content ← tokenize text received from Split_content
Retained_kw ← filter Token_content using classified keyword list
Store retained keword and URL
If loop reaches desired number of URLs
exit loop
else
continue
end if
Step 3.end loop

Figure 2. Classification algorithm

In the next section a snapshot of the raw text which is the text in its original containing a lot of noise data is presented. This is shown in figure 3. This raw text needs a lot of cleaning process which is done afterwards it is fetched by reading the URL's.

b'\t\t\t NOTES\n\t\t\t =====\n\n\t\t JavaMail(TM) API 1.6.1 release\n\t\t ----------\n\nWelcome to the 1.6.1 release of the JavaMail API implementation. \n\nPlease refer to CHANGES.txt for a list of the changes since the \nprevious release.\n\nPlease see the FAQ at https://javaee.github.io/javamail/FAQ\n\nProtocol Providers\n----------\n\nThe JavaMail API jar file "mail.jar" includes the full JavaMail API\nimplementation and the Sun protocol providers - IMAP, SMTP, and\nPOP3. The simplest way to use the JavaMail API is to just use the\nmail.jar file and ignore the other jar files in this package.\n\nIn some cases it may be desirable to minimize the size of the JavaMail\nAPI code used by an application (e.g., when downloading with an applet).\nIn this case you might want to include the "mailapi.jar" file, which\nincludes *no* protocol providers, along with just the jar file for the\nprotocol provider you need. For example, an applet that only needs to\nsend mail could use the "mailapi.jar" file and the "smtp.jar" file.\n\nAn important note when using the separate protocol provider jar files:\n\n- You can\'t mix and match the Sun protocol providers between different\n releases of the JavaMail API. The Sun protocol providers depend on\n implementation-specific utility APIs within the mailapi.jar file.\n (Third party protocol providers that don\'t depend on these APIs\n should work fine.)\n\n\nNOTE: The Sun protocol provider documentation is available in javadoc format,\n see docs/javadocs/index.html in the directory where you extracted\n the JavaMail API zip file. This documentation describes how to\n use features of the Sun protocol providers to directly access\n some features of the SMTP, IMAP, and POP3 protocols that are\n not otherwise supported by the standard JavaMail API.\n\n\nGmail IMAP Provider\n----------\n\nThis release includes an EXPERIMENTAL Gmail IMAP provider.\nNormal use of Gmail is handled by the standard "imap" protocol\nprovider, but the new "gimap" protocol provider supports additional\nGmail-specific non-standard features. See the javadocs for the\ncom.sun.mail.gimap package for details. Note that the gimap.jar file\nneeds to be added to your CLASSPATH to use this new provider.\n\n\nSASL Support\n----------\n\nOn systems that support the Java SASL API (javax.security.sasl, JSR-28),\nsuch as J2SE 5.0 and later, the IMAP provider can use the SASL API to\nfind an appropriate authentication mechanism. The SASL API also allows\nyou to plug in support for custom authentication mechanisms. See The\nJava SASL API Programming and Deployment Guide SASL support.\n\n\nDSN Support\n----------\n\nThis release of JavaMail includes EXPERIMENTAL support for creating\nand parsing Delivery Status Notifications, as defined by RFC 3462\nand RFC 3464. To make use of this support you need to include dsn.jar\nin your CLASSPATH along with mail.jar. See the javadocs for the\ncom.sun.mail.dsn package for more details.\n\nThe DSN package also provides support for creating and parsing Message\nDisposition Notifications, as defined by RFC 3798.\n\nThe APIs unique to this package should be considered EXPERIMENTAL.\nThey may be changed in the future in ways that are incompatible with\napplications using the current APIs.\n\n\nNTLM Support\n----------\n\nThis release of JavaMail includes EXPERIMENTAL support for the\nMicrosoft NTLM authentication mechanism used by Exchange. See the\nfile NTLMNOTES.txt for details.\n\n\nOSGi Support\n----------\n\nThe JavaMail jar files are now OSGi bundles. Please let us know\nof any problems using JavaMail with OSGi.\n\n\nHow to submit bug reports\n----------\n\nIf you\'ve found a bug, or if you just need help figuring out how to use\nthe JavaMail API, please try to include the following information in\nyour message to us:\n\n - a program or code snippet that

Figure 3. Raw Text Read from URL

In figure 4 raw texts is converted into clean text by applying text mining cleaning techniques. This converts the whole text into more understandable format as well as also reduces the overall size of the text.

JavaMailTM API releasenttnnWelcome release of JavaMail API implementation nnPlease refer CHANGEStxt for list of changes since npreviousreleasennPlease see FAQ ProvidersnnnTheJavaMail API jar file mailjar includes full JavaMailAPInimplementation Sun protocol providers IMAP SMTP andnPOP3 simplest way use JavaMail API just use thenmailjar file ignore other jar files this packagennIn some cases may desirable minimize size of JavaMailnAPI code used by application eg downloading with appletnIn this case you might want include mailapijar file whichnincludes no protocol providers along with just jar file for thenprotocol provider you need For example applet that only needs tonsend mail use mailapijar file smtpjarfilennAn important note using separate protocol provider jar filesnn You cant mix match Sun protocol providers between differentn releases of JavaMail API Sun protocol providers depend onnimplementationspecific utility APIs within mailapijarfilen Third party protocol providers

Figure 4. Clean text

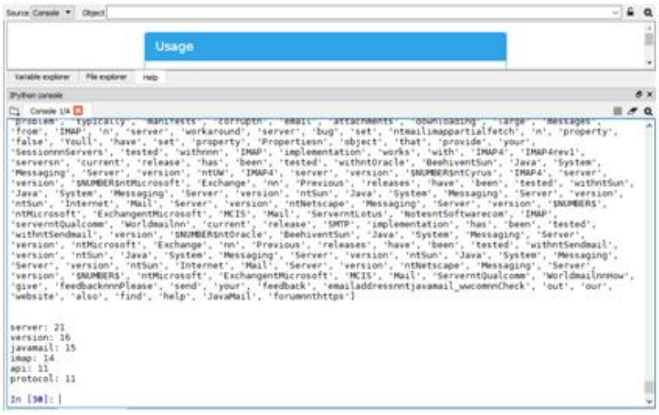

Figure 5. Retain Keywords

The output of keywords which are retained are 12 in number and are presented below along with returned words, retained words and retained keywords.

python: 10
online: 3
free: 2
recommended: 2
python3: 2
programming: 2
code: 2
elanations: 2
fun: 2
learning: 2
web: 2
module: 2

Returned words = 2732
Retained words = 142
Retained KW = 12

In the next part table 1 shows the list of URL's related to E-learning and in table 2 a count of URL's along with how much of the word is read from those URL's and then word filtered and finally set of keywords retained is presented

Table 1. List of URL'S

S.No.	URL'S
1.	https://javaee.github.io/javamail/docs/NOTES.txt
2.	https://gist.githubusercontent.com/pwicherski/43f39516a523f18df0e15d6bbc8b0bc7/raw/87147fc6b2e11b3be375ee9a7e334c90eb0f9e07/Python.txt
3.	http://cyclismo.org/tutorial/R/
4.	https://www.pythoncourse.eu/python3_course.php
5.	https://www.youtube.com/watch?v=3u1fu6f8Hto
6.	https://www.rubylang.org/en/documentation/quickstart/
7.	https://golangbot.com/
8.	https://www.kdnuggets.com/2016/03/datacamp-r-learning-path-7-steps.html
9.	http://www.jython.org/docs/tutorial/indexprogress.html

Table 2. URL'S and Retain keywords from them

URL	Word Read from URL	Word after filtering	Kew words retained
1	9542	828	10
2	2732	142	12
3	12841	729	7
4	16687	803	8
5	18972	176	12
6	17346	114	14
7	16124	123	10
8	19382	156	9
9	18574	121	11

IV. PERFORMANCE ANALYSIS

To analyse the performance of the proposed work precision, recall and accuracy were calculated on the given criterion and the statistical formulas is calculated as follows.

Criterion
True Positive (TP)←denotes the correctly identified key words.
False Positive(FP) ← denotes keywords incorrectly identified as other words
False Negative(FN)← Incorrectly identified as key words.
True Negative(TN)← Correctly rejected words.

True Positive Rate(TPR) or Sensitivity is calculated using eq(6) which is the sum of true positive and false negative
TPR ←(TP + FN) ---------- (6)

True Negative Rate(TNR) or Specificity is calculated using eq(7) which includes the addition of true negative with the false positive
TNR ←(TN + FP) ---------- (7)

Accuracy can be calculated by using the formulae in equation 8
Acc = (TP + TN)/ (TP + FP + TN + FN)

Table 3. Total Words Retained, Correct Words and Irrelevant words from various URL'S

URL	TP	FP	TN	FN
1	10	0	9532	0
2	12	0	2720	0
3	7	2	12831	1
4	8	63	16615	1
5	12	176	18784	0
6	14	52	17279	1
7	10	32	16080	2
8	9	114	19258	1
9	11	86	18477	0

In table 3 the retained words were filtered out using the proposed text mining technique to arrive at values of TP, FP, TN, and FN. FP contain the words which were not found to be standard technical words. The rejected words (TN) were calculated after browsing the total words retained from the URL. The retained clean file was browsed to check correctly rejected words and incorrectly rejected words and the count was noted.
Similarly in order to measure the performance precision and recall are the two measures that have been used.

Precision: It is also known as measure of correctness and is defined as the fraction of relevant documents among the retrieved documents.
Precision= correctly retrieved words/total number of words
Another measure that have been used is recall which is also known as measure for describing the completeness and is defined as the relevant documents which are successfully retrieved or number of correct results divided by variety of results that ought to returned.
Recall= correctly retrieved documents/ total of relevant documents retrieved.
The results of precision, recall and accuracy is shown in the table 4 below.

Table 4. Results of Precision, Recall and Accuracy Obtained from Table 3

S.No	Precision	Recall	Accuracy
1	1	1	1
2	1	1	1
3	0.777778	0.875	0.999766
4	0.112676	0.888889	0.996165
5	0.06383	1	0.990723
6	0.212121	0.933333	0.996945
7	0.238095	0.833333	0.997891
8	0.073171	0.9	0.994067
9	0.113402	1	0.99537

Next table 5 shows the URL'S and the associated keywords with them which are retained and the class with which they belong to. This leads to the development of a classified repository which consists of the URL'S, most retained keywords and their class with which they are related to

Table 5. URL'S Retained with Keywords and Class

URL	Key words	Class
https://javaee.github.io/javamail/docs/NOTES.txt	Server, version, javamail, imap, api, protocol, support, java, exchange, messaging	Java
https://gist.githubusercontent.com/pwicherski/43f39516a523f18df0e15d6bbc8b0bc7/raw/87147fc6b2e11b3be375ee9a7e334c90eb0f9e07/	Python, online, free, recommended, python3, programming, code, elanations, fun, learning, web, module	Python

http://cyclismo.org/tutorial/R/	Class reference, internal, class to ctreel1a, nn, basic, r, script	R

V. CONCLUSION AND FUTURE SCOPE

From the analysis of the results it can be concluded that the purpose of the proposed work has been achieved successfully. On an average the retained keywords were 8, Precision, Recall and Accuracy were observed at 0.964, 0.982 and 0.97 respectively which supports the robustness of the proposed classification algorithm. The authors propose to further enhance the proposed algorithm and use the classified model to design a machine learning based prediction model for giving accurate suggestions to the surfers about related e-learning sites based on their search.

For the future work the pre-processed and classified web data would be used for applying machine learning algorithm and predictive modelling.

REFERENCES

[1] J.Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, "*Recommender Systems Survey*", Knowledge-based systems, Vol 46, pp.109-132, 2013.

[2] X. Zhou, Y. Xu, Y. Li, A. Josang, C. Cox, "*The State-of-the-Art in Personalized Recommender Systems for Social Networking*", Artificial Intelligence Review, Vol37, Issue2, pp.119-132, 2012.

[3] S. Kannan, V. Gurusamy, "*Preprocessing Techniques for Text Mining*" 2014.

[4] W. Kuang, N. Luo, Z. Sun, "*Resource Recommendation based on Topic Model for Educational System*", In Information Technology and Artificial Intelligence Conference (ITAIC 2011), Vol. 2, pp. 370-374,2011.

[5] J.Xiao, M. Wang, B. Jiang, J. Li, "*A Personalized Recommendation System with Combinational Algorithm for Online Learning*", Journal of Ambient Intelligence and Humanized Computing, pp.1-11, 2017.

[6] J.K.Tarus, Z.Niu, D. Kalui, " *A Hybrid Recommender System for e-learning based on Context Awareness and Sequential Pattern Mining*", Soft Computing, pp.1-13, 2017.

[7] P.Do, K. Nguyen, T.N. Vu, T.N. Dung, T.D. Le, "*Integrating Knowledge-Based Reasoning Algorithms and Collaborative Filtering into E-Learning Material Recommendation System*", In International Conference on Future Data and Security Engineering, Springer, Cham, pp. 419-432, 2017.

[8] T.Y.Tang, G. McCalla, "*Smart Recommendation for an Evolving e-learning System: Architecture and Experiment*", International Journal on elearning, Vol4 issue1, pp.105, 2005.

[9] R.J.Mooney, L. Roy, "*Content-based Book Recommending using Learning for Text Categorization*", In Proceedings of the fifth ACM conference on Digital libraries, pp. 195-204, 2000.

[10] A.Al-Hamad, N, Yaacob, A.Y. Al-Zoubi, "*Integrating 'Learning Style'Information into Personalized E-learning System*", IEEE Multidisciplinary Engineering Education Magazine, Vol3 Issue1, pp.2-6 ,2008.

[11] L.Zhuhadar, O. Nasraoui, R. Wyatt, E. Romero, "*Multi-model Ontology-based Hybrid Recommender System in E-learning Domain*", In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences , Vol. 3, pp. 91-95, 2009 .

[12] S. Fraihat, Q. Shambour, "*A Framework of Semantic Recommender System for e-learning*". Journal of Software, Vol10 Issue 3, pp.317-330, 2015.

[13] C. Yang, J.Hung, H.Hsu, " *A Web Content Suggestion System for Distance learning*", September 2006.

[14] M.K.Khribi, M. Jemni, O. Nasraoui, "*Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval*" , In Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on ,pp. 241-245, 2008.

[15] G.Lv, C. Hu, S. Chen, S., " *Research on Recommender System based on Ontology and Genetic Algorithm*". Neurocomputing, 187, pp.92-97, 2016.

[16] P. Dwivedi, K. K. Bhardwaj, " *E-learning Recommender System for a Group of Learner based on the Unified Learner Profile Approach*", Expert Systems, pp. 264-276, 2015.

Authors Profile

Ms. Kamika Chaudhary pursued bachelor of technology and master of technology in computer science from Uttar Pradesh Technical University Lucknow. Currently she is a research scholar in department of computer science in Gurukul Kangri Vishwavidyalaya Campus in dehradun, India. She has published research papers in scopus indexed journal and has also attended international conferences.She has research interest in the field of web mining, information retrieval.

Dr. Neena Gupta is working as an assistant professor in department of computer science of Kanya Gurukul Campus (Gurukul Kangri Vishwavidyalaya) Dehradun, India.She has more than 11 years of experience and has guided several Ph.D. research scholars. Her research interest includes distributed database, data mining and web database. She has published a large number of research papers in international journal of repute and has attended many national/international conferences.

