International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 4 Issue 6, June 2015

Feature Recommendation from Competitive Product Description by using KNN with Different Similarity Formulas

Ms. Syeda Nazema Syed Subhan, S. N. Deshmukh

***Abstract*— Domain analysis is useful in many applications for finding their similar and dissimilar parts. Many software applications include generally domain analysis activities. Recommendation Systems are techniques providing suggestions for items to be of use to a user. Much software includes extensive recommendation techniques. In this paper we present a recommendation system that is design for feature recommendation on the basis of already existing products. We used K-Nearest Neighbor machine learning technique for recommending features by using four different Distance formula i.e. Cosine Similarity, Jaccard Similarity Coefficient, Correlation and Hamming Distance formula during domain analysis. Our feature recommendation algorithm is quantitatively evaluated and the results are presented. Furthermore the performance of the recommender system is demonstrate and evaluated within the context of experiment. The results clearly highlight the benefits of our approach.**

***Index Terms* — Domain Analysis, Recommender Systems, k – Nearest Neighbor (kNN), Cosine Similarity, Jaccard Similarity Coefficient, Correlation, Hamming Distance.**

I. INTRODUCTION

DOMAIN analysis is the process of identifying, organizing, analyzing, and modeling features common to a particular Domain [1], [2]. It is conducted at the starting phase of software development life cycle. Recommendation Systems are techniques providing suggestions for items to be of use to a user. Recommender systems have become extremely common in many software applications; it is useful in a variety of applications. Mostly used in movies, books, music, news, social tags, research articles and online products in general. Some domain analysis techniques are used such as the Feature – Oriented Domain Analysis (FODA) [2] or the Feature – Oriented Reuse Method (FORM) [3] to perform successfully these techniques it is reliant on the obtain ability of related brochures, entrance to the competitive project repositories and the knowledge of the domain analyst. Other technique such as the Domain Analysis and Reuse Environment (DARE) [4] use data mining

***Manuscript received June, 2015*.**

Syeda Nazema Syed Subhan, *Dept. of CS & IT, Dr. Babasaheb Ambedkar Marathwada University Aurangabad, India, 9923613786.*
S. N. Deshmukh, *Dept. of CS & IT, Dr. Babasaheb Ambedkar Marathwada University Aurangabad, India.*

and information retrieval methods to deliver robotic support for feature identification and extraction. Unsupervised association rule mining technique such as Apriori algorithm [5], [6], FP_growth algorithm, k-Nearest Neighbor Approach common in collaborative filtering recommender systems [8] is used. But the unsupervised association rule mining technique apply when user provide only few feature then Binary k-Nearest Neighbor Approach[9],[10] apply. The products with less than six features were ignored, as their profiles are too sparse to create good neighborhoods but when any person want to know the features in particular domain and they don't have knowledge of particular domain then this recommendation system is not helpful for them, because in above feature recommendation system it is necessary for user to give greater or equal to six feature then the system recommends them the other features. The recommendation of features is therefore limited by the scope of the available product specifications.

In this paper, we address these limitations through presenting a novel approach for recommending features, no restriction of product specification. Here we used K – Nearest Neighbor (kNN) classification approach with four different similarity formulas for recommending features. Our approach has two options. The first one is where the user has product description, then we take input as initial product description then analyze this description, and recommending features based on the provided description by using K-Nearest Neighbor (kNN) classification approach which is very common in Collaborative Filtering Recommendation System on the basis of competitive products.

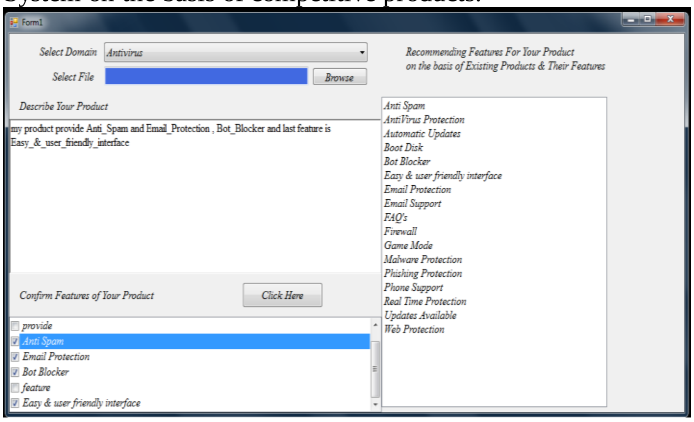

Figure 1: Example of Feature Recommendation when user provides description.

 All Rights Reserved © 2015 IJARCET

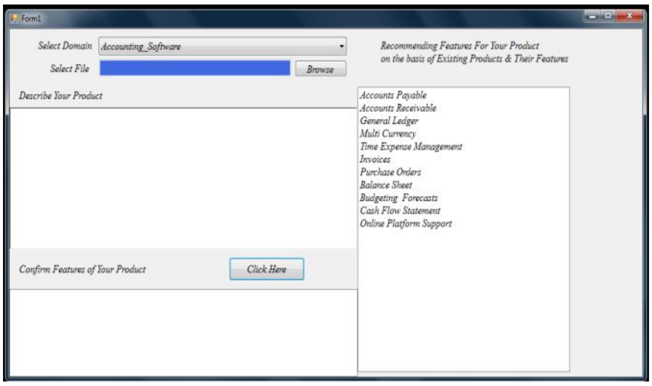

Figure 2: Example of Feature Recommendation when user does not provide any description.

The second one is where the user of the recommendation system doesn't have any feature description available with them, and wants to know the features for a particular domain, and then also we provided them the feature recommendations. Figure 1 illustrates a feature recommendation scenario for antivirus product when user provide sufficient description of their product and figure 2 illustrates a feature recommendation scenario for antivirus product when user don't provide any description .

II. Overview

Our feature recommendation system includes following steps. According to the user selection, if they select first option i.e. user is having product description then it recommends features in following steps as illustrated in Fig 3.

First user provide initial product description then in second step we performed Natural Language Processors (NLP) after this we applied kNN on data by using Cosine Similarity, Jaccard Similarity Coefficient, Correlation and Hamming Distance formula. It recommends features if user want more recommendation then again we applied kNN. Fig 4 shows the recommendation system if user does not have any initial product description then recommendation system recommends the features which has high priority in dataset.

III. Raw Product Data

The entire feature descriptors used in this paper was mined from Softpedia.com, Findthebest.com, Google Apps, and

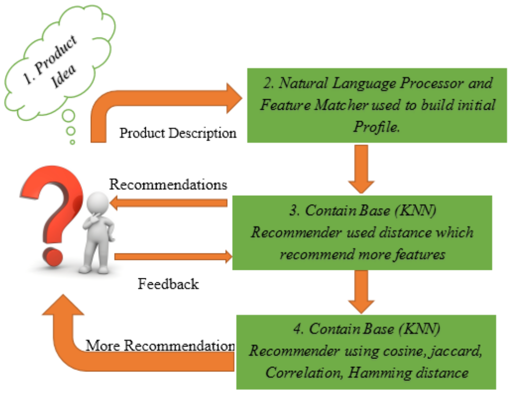

Fig 3: Recommender System to recommend Feature if user has initial product description

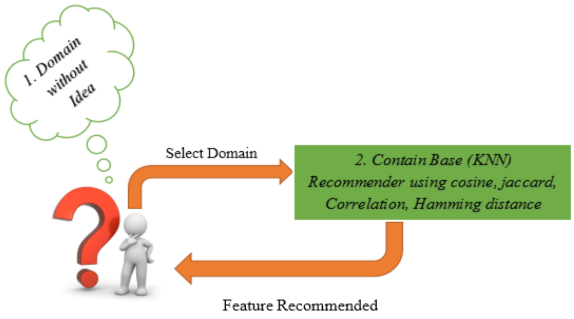

Figure 4: Recommender System to recommending feature if user does not have the initial product description.

Softonic.com. However, our approach is easily adapted for use with product descriptions from other online sources Softpedia.com, findthebest.com provides descriptions for an extensive variety of software products including Windows, Linux, Mac, Mobile, and Web applications. The incremental diffusive clustering algorithm [11], [12], [13], [14] is used in some recommendation system for extracting product feature descriptors into an aggregate representation of a candidate feature. Most products include a bullet-point list format section of feature descriptors as in softpedia.com. Our data set contains 42 different domains as show in table 1. Table2 illustrate a small subsection of the feature manually recognized from Softpedia.com Antivirus software products.

Table 1: Product Categories

Product Category	Prod. Count	Feature Count
Accounting Software	192	41
Animation Software	74	72
Antivirus	240	35
Audio Editing Software	160	90
Billing Invoicing Software	224	39
Business Intelligence Software	175	24
CAD Software	57	70
Call Center Software	150	50
Computerized Maintenance Mgmt System	49	127
Construction Management Software	243	30
Contact Management Software	44	33
Contract Management Software	130	28
Database Management Systems	60	28
Distribution Software	113	32
Endpoint Protection Software	126	19
Facility Mgmt Software	46	10
Field Service Management	174	12
Fleet Management Software	249	26
Fundraising Software	139	34
Help Desk Software	204	25
Hotel and Hospitality Mgmt Software	187	19
HRM Software	140	15
Inventory Software	133	25
Legal Software	139	28
Marketing Automation Software	132	20
Network Mgmt Software	99	40
Payroll Software	57	20
Project Management Software	450	40
Media Players	97	60
Medical Software	8	35
Video Editing Software	122	62
Web Browser	51	52
Web Design Software	118	56

Table 2: A Sample of Features Collected from Softpedia Antivirus Products Description

	a-squared Free	Acronis AV	AhnLab Platinum	Ahnlab	Anti-Trojan Elite	AppRanger	Ashampoo a.m.	Auslogics AV	Avast! Pro AV	Avast! Int. Sec	AVG AV Pro.	AVGAV+Firewall	Avira AntiVir Premium	Avira SmallBusiness Suite	Bkav2008	BitDefender Total Sec. '10	BitDefender Int. Sec. '10	Corbitek AntiMW	CyberDefender Int. Sec.	COMODO Cloud Scanner	Dr.Web	G DATA AV '10	Fix-it Utilities Pro.
Active detection of downloaded files		●	●					●	●		●	●	●	●		●	●					●	
Active detection of instant messenging		●	●	●				●	●		●	●				●	●					●	
Active detection of removable media									●		●	●											●
Active detection of search results		●	●						●		●	●	●	●		●	●					●	
Anti-Root kit scan		●				●			●	●	●	●	●	●	●					●		●	●
Automatic scan									●				●	●	●	●	●						
Automatic scan of all files on startup									●					●	●							●	●
Automatic updates					●				●		●	●	●	●	●	●	●	●	●		●	●	
Behavioral Detection									●					●	●								
Command line scan									●	●													
Contain viruses in specific quarantine					●				●					●	●	●	●		●				
Customized firewall settings										●		●											
Data encryption		●	●	●				●	●		●	●				●	●						
Detection of suspcious/injected DLLs																							
Disc scan for finding malware	●	●	●	●	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●
Email detection	●	●	●						●		●	●	●	●		●	●		●		●	●	●
File backup																●	●						●
...																							
...																							

	Gucup AV	GSA Delphi Induc Cleaner	Hazard Shield	GGreat Owl USBAV	Jiangmin AV KV '10	K7 AV	Immunet Protect	MW Destroyer	Kaspersky Ultra-Portables	Mx One AV	MultiCore AV AntiSpyware	McAfee VirusScan	Microworld AV Toolkit Util.	Norton Int. Sec.	Novashield - Anti MW	NoVirusThanks MW Rem.	Norman Virus Control	Norton AV 2009 GamingEd.	Norton AV	Norman Sec. Suite	Network MW Cleaner	PC Tools Int. Sec. '10	Outpost Sec. Suite Pro
Active detection of downloaded files						●	●					●		●			●		●	●	●	●	●
Active detection of instant messenging														●					●	●			●
Active detection of removable media				●						●			●	●						●			●
Active detection of search results												●		●					●	●			●
Anti-Root kit scan	●				●		●					●		●	●	●			●	●	●		●
Automatic scan	●		●																●			●	
Automatic scan of all files on startup	●				●									●		●					●	●	●
Automatic updates			●		●	●	●			●		●		●		●	●		●		●	●	●
Behavioral Detection					●							●		●	●								
Command line scan																							
Contain viruses in specific quarantine														●						●			●
Customized firewall settings																							●
Data encryption														●									●
Detection of suspcious/injected DLLs																							
Disc scan for finding malware	●	●	●		●	●	●	●		●	●	●	●	●	●	●	●		●	●	●	●	●
Email detection					●	●						●		●			●		●	●			●
File backup		●												●		●							
...																							
...																							

	nProtect GameGuard Pers.	Quick Heal Int. Sec. '10	Quick Heal Total Sec. '10	Steganos Int. Sec. 2009	Steganos AV 2009	SpyDLLRem.	Tizer Rootkit Razor	The Shield Deluxe '10	The Cleaner 2011	SystemSuite Pro.	SysIntegrity AM	VIPRE AV Premium	VirIT eXplorer Lite	TrustPort PC Sec. '10	Twister Anti-TrojanVirus	ZoneAlarm Sec. Suite	Wlording AntiSpyware	Your Free AntiSpyware	Webroot Int. Sec.	Windows AV Mate Prof.	Wimp	VIRUSfighter	VirusBuster Pro.	VirusBuster Personal
Active detection of downloaded files		●						●				●			●	●			●					
Active detection of instant messenging		●						●				●			●				●					
Active detection of removable media		●	●								●	●							●					
Active detection of search results		●						●				●				●					●			
Anti-Root kit scan		●	●		●	●	●	●			●	●	●		●	●			●	●	●			
Automatic scan											●	●	●	●	●	●			●	●			●	●
Automatic scan of all files on startup										●		●		●						●			●	●
Automatic updates	●			●				●	●	●		●	●		●	●	●		●	●		●	●	●
Behavioral Detection				●								●			●									
Command line scan									●															
Contain viruses in specific quarantine				●					●		●				●			●	●				●	●
Customized firewall settings												●												
Data encryption	●													●		●							●	
Detection of suspcious/injected DLLs						●																		
Disc scan for finding malware	●	●	●	●	●		●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Email detection		●		●	●			●				●			●	●			●				●	
File backup																			●					
...																								
...																								

It displays that the most of the examined Anti-virus systems includes core features such as Antivirus Protection, Web Protection, Automatic Updates, as well as deviations such as Key loggers, File Shredding, Game Mode which are found in only a small subset of products. Moreover, certain features such as AntiSpyWare/Adware are found primarily in antivirus software applications, while others, such as Easy and User Friendly Interface are common across many different product categories. Various associations can be found between features.

IV. Feature recommendation by using Standard KNN

After feature have been identified data store in the form of a product by feature matrix [15][16] i.e. $D = \left(d_{i,j}\right)_{m\times f}$ here p represent the number of products available in particular domain and f represent number of features available in particular domain. $d_{i,j}$ Is equal to 1 it means feature j includes in i product description and if $d_{i,j}$ is equal to 0 then it means feature j not includes in i product description. The complete set of recommendable feature is considered as feature pool from now for generating feature recommendation.

a. Creating new primary product Description

Our approach takes input as an initial textual description of new product. First performed the basic preprocessing such as tokenization, removed unwanted terms (stop word), stemming. The Cosine Similarity is used here to match individual segment to feature in feature pool. If the product description might describe a feature which is not matched to the feature pool. This may occur because the feature pool does not provide analysis of the proposed functionality. Or a match may not be found because the analyst had not defined the product in enough detail or used wording not characterized in the model. In this situation, the recommender system will recommend same features when user doesn't have product description. If user doesn't have any description of product then there's no need to do preprocessing.

b. Feature Recommendation using kNN with Cosine Similarity

As we study in previous paper the K Nearest Neighbor algorithm is efficient for feature recommendation [17], [18], [19].For determinations of feature recommendations, the similarity of the new product and each of the competitive products in the dataset is calculated by using Cosine Similarity formula and the top k (10) most similar products are consider as neighbors of the new product. We established a threshold score is 0.60 for Cosine Similarity. Formula used for Cosine Similarity between new product *np* and existing product *ep*, *prodctsim (ep,np)* is given below:

$$\Pr oductSim(ep,en) = \frac{F_{ep} \cap F_{np}}{\sqrt{\left|F_{ep}\right| \bullet \left|F_{np}\right|}}$$

Where F_{ep} denotes set of features of product p[32] and F_{np} denotes a set of features of n new products. Product similarity is in between 0 and 1. If it is 1 then it means both subset of feature are similar, 0 both subset are dissimilar [21], [22]. An important property of the Cosine Similarity is its independence of document length. For example, combining two same copies of a document *d* to get a new document *d0*, the Cosine Similarity between *d* and *d0* is 1, which means that these two documents are similar.

c. Feature Recommendation using kNN with Jaccard Similarity Coefficient

The Jaccard Coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the two document *d1 = d2* and it is 0 when $d1 \neq d2$ where 1 means that the two objects are the same and 0 means that they are completely different. Here we established a threshold score of 0.6, formula used for finding Jaccard Similarity between new product *np* and existing product *ep*, *prodctsim (ep,np)* is given below:

$$\Pr oductSim(ep,en) = \frac{F_{ep} \cap F_{np}}{F_{ep} \cup F_{np}}$$

Where F_{ep} denotes a set of features of product p[32] and F_{np} denotes a set of features of n new products. Product similarity is in between 0 and 1.Here the top k (10) most similar products are considered as neighbors of the new product [22].

d. Feature Recommendation using kNN with Correlation

Correlation gives any statistical relationship between sets of data. Correlations are useful because they can indicate a predictive relationship that can be exploited in practice [23]. In correlation we established a threshold score of 0.6,

 All Rights Reserved © 2015 IJARCET

formula used for finding correlation between new product *np* and existing product *ep*, *prodctcorr (ep,np)* is given below:

$$ProductCorr(ep,np) = \left[\frac{Covariance(F_{ep},F_{np})}{standarddeviation(F_{ep})*standarddeviation(F_{np})}\right]$$

Where F_{ep} denotes a set of features of product p [32] and F_{np} denotes a set of features of n new products.

Product correlation is in between -1 and 1. It is -1 when the two document $d1 \approx d2$, it is 0 when $d1 \neq d2$ and is 1 when *d1= d2*, where 1 means the two objects are the same and 0 means they are completely different, -1 means *d1* is completely opposite to *d2*. Here we used the top k (10) most similar products are considered as neighbors of the new product.

e. Feature Recommendation using kNN with Hamming Distance formula

The last one formula for finding similarity between new product and existing product used in KNN is Hamming Distance formula. Hamming finds the number of features that are different between new product feature set and existing product features set. If the number of features that are different is less than new feature set then we can say both feature sets have some features similar, if the distance is 0 means both feature sets are similar. If the number of features that are different is equal to total number of feature in new product feature set then we can say new product and existing product are completely opposite[25], [26] .Formula used for finding Hamming Distance between new product *np* and existing product *ep*, *prodctdist(ep,np)* is given below:

Productdist(ep,np)= Number of different bit's when comparing *ep,np*

V. Feature Recommender Evaluation

a. Evaluation Metrics

We use precision, recall, and Accuracy as our metrics to evaluate the performances of the methods. Precision and recall are defined as follows:

Precision (*P*) is the percentage of positive predictions those are correct.

$$P = TP / (TP + FP)$$

Recall (*R*) is the percentage of positive labeled instances that were predicted as positive.

$$R = TP / (TP + FN)$$

Accuracy is the percentage of predictions those are correct.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Here:

TP (True Positives) is the number of features classified correctly as positive;

Table 3: The Precision for the Different Distances.

Precision	Cosine	Jaccard	Correlation	Hamming
Antivirus	0.63008	0.63008	0.61994	0.61476
Animation	0.72605	0.72605	0.7421	0.72605
Audio Editing	0.65232	0.65232	0.50116	0.65232
Business Intelligence	0.52332	0.52332	0.64368	0.65892

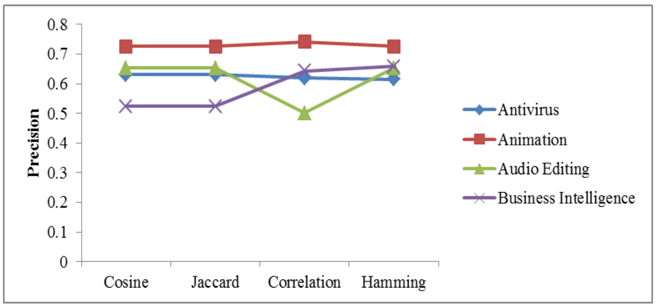

Fig 5: Graph of Precision value for the different distances

FP (False Positives) is the number of negative features that are classified as positive incorrectly by the classifier;
TN (True Negatives) is the number of negative features that are classified as negative correctly by the classifier.
FN (False Negatives) is the number of positive features that are classified as negative incorrectly by the classifier. [28]

b. Analysis of result

Tables: 3, 4, 5 show the results from the experimental evaluation of the kNN Classification with the different distance measures. In the following, we briefly discuss the results for each evaluation measure.

The precision and recall have inverted values. In the case of precision, hamming distance is the best performing, as shows in figure 5.While for recall it is the worst performing compared to correlation distance as shows in Fig 6. The distance to the correlation methods is large in the both cases. This means that the labels produced with hamming distance are reliable (low false positive rate); however, they do not cover all relevant labels for a given example (high false negative rate). The other two i.e. cosine similarity, Jaccard similarity have similar performances to each other.

When comparing performance we come to know that correlation is having good performance then other distance formulas. The hamming distance has the lowest performance (because of the weak results for recall) but it is slightly better than the remaining two distances.

Table 4: The Recall for the Different Distances

Recall	Cosine	Jaccard	Correlation	Hamming
Antivirus	0.46436	0.46436	0.776	0.56824
Animation	0.599375	0.599375	0.521175	0.599375
Audio Editing	0.60816	0.60816	0.41144	0.60816
Business Intelligence	0.40386	0.40386	0.5257	0.54244

.

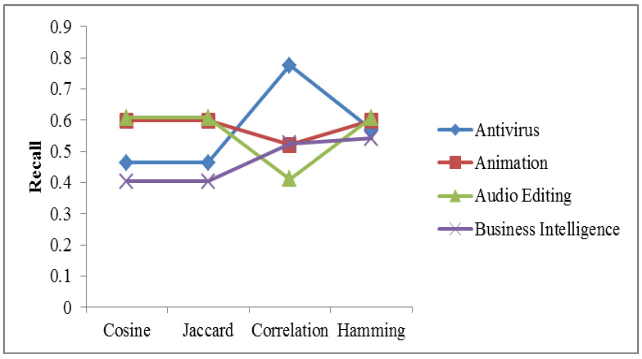

Fig 6: Graph of Recall value for the Different Distances.

Table 5: The Average Accuracy for the Different Distances.

Accuracy	Cosine	Jaccard	Correlation	Hamming
Antivirus	45.46	45.46	59.292	49.556
Animation	54.2175	54.2175	50.6325	54.2175
Audio Editing	51.488	51.488	36.806	51.488
Business Intelligence	41.488	41.488	51.296	53.162

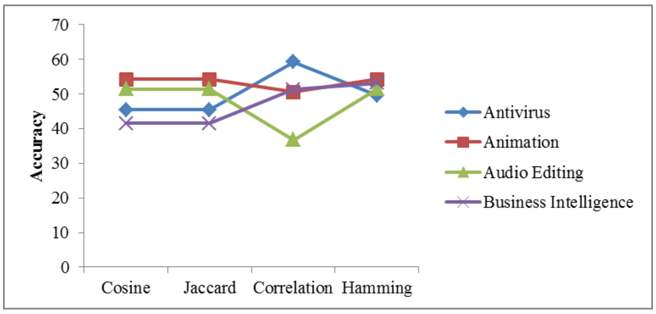

Fig 7: Graph of Accuracy for the Different Distances.

VI. Conclusion

We have conducted several experiments using number of initial keyword as shown above in the table 3, 4 and 5. From the table 3 and Fig 5 it is clearly visible that best precision values were obtained using the Hamming Distance Formula followed by Jaccard, Cosine and Correlation. We can't guess the performance only on the basis of result based on Precision. We considered here the results of Recall and Accuracy, and then we came to know that Correlation having the better Recall as compared to others and its Accuracy is also highest. Hence recommending features using kNN classification with Correlation formula gives better performance, more number of feature recommendations as compared to others.

Acknowledgment

This experiment has been partially supported by Department of Computer Science and IT Dr. Babasaheb Ambedkar Marathwada University, Aurangabad. The views expressed here are those of the authors only.

REFERENCES

[1] G. Arango and R. Prieto-Diaz, Domain Analysis: Acquisition of Reusable Information for Software Construction. IEEE CS Press May 1989.

[2] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Technical Report CMU/SEI-90-TR-021, Software Eng. Inst., 1990

[3] K.C. Kang, S. Kim, J. Lee, K. Kim, G.J. Kim, and E. Shin, "FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures," Annals of Software Eng., vol. 5, pp. 143-168, 1998.

[4] W. Frakes, R. Prieto-Diaz, and C. Fox, "Dare: Domain Analysis and Reuse Environment," Annals of Software Eng., vol. 5, pp. 125- 141, 1998.

[5] Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases, 1994.

[6] R. Agrwal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), pp. 487-499, Sept. 1994.

[7] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00), June 2000.

[8] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," The Adaptive Web, pp. 291-324,Springer, 2007.

[9] C. Castro-Herrera, J. Cleland-Huang, and B. Mobasher, "Enhancing Stakeholder Profiles to Improve Recommendations in Online Requirements Elicitation," Proc. IEEE Int'l Conf. Requirements Eng., pp. 37-46, 2009.

[10] C. Castro-Herrera, C. Duan, J. Cleland-Huang, and B. Mobasher, "A Recommender System for Requirements Elicitation in Large-Scale Software Projects," Proc. ACM Symp. Applied Computing,pp. 1419-1426, 2009.

[11] H. Dumitru, M. Gibiec, N. Hariri, J. Cleland-Huang, B. Mobasher, C. Castro-Herrera, and M. Mirakhorli, "On-Demand Feature Recommendations Derived from Mining Public Software Repositories," Proc. 33rd Int'l Conf. Software Eng., p. 10, May 2011.

[12] C. Duan, J. Cleland-Huang, and B. Mobasher, "A Consensus Based Approach to Constrained Clustering of Software Requirements," Proc. 17th ACM Conf. Information and Knowledge Management, pp. 1073-1082, 2008.

[13] J. Cleland-Huang and B. Mobasher, "Automated Detection of Recurring Faults in Problem Anomoly Reports," SERC Report to Lockheed Martin, 2009.

[14] Negar Hariri, Carlos Castro-Herrera, Member, IEEE, Mehdi Mirakhorli, Student Member, IEEE, Jane Cleland-Huang, Member, IEEE, and BamshadMobasher, Member, IEEE "Supporting Domain Analysis through Mining and Recommending Features from Online Product Listings"IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 12, DECEMBER 2013.

[15] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective Personalization Based on Association Rule Discovery from Web Usage Data," Proc. Third Int'l Workshop Web Information and Data Management (WIDM '01), Nov. 2001.

[16] J. Sandvig, B. Mobasher, and R. Burke, "Robustness of Collaborative Recommendation Based on Association Rule Mining," Proc.ACM Conf. Recommender Systems, 2007.

[17] Syeda Nazema Syed. Subhan, S. N. Deshmukh "A Survey on Feature Recommendation Techniques" International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169Volume: 3 Issue: 3.

[18] C. Castro-Herrera, J. Cleland-Huang, and B. Mobasher, "Enhancing Stakeholder Profiles to Improve Recommendations in Online Requirements Elicitation," Proc. IEEE Int'l Conf. Requirements Eng., pp. 37-46, 2009.

[19] C. Castro-Herrera, C. Duan, J. Cleland-Huang, and B. Mobasher, "A Recommender System for Requirements Elicitation in Large-Scale Software Projects," Proc. ACM Symp. Applied Computing, pp. 1419-1426, 2009.

[20] E. Spertus, M. Sahami, and O. Buyukkokten, "Evaluating Similarity Measures: A Large-Scale Study in the Orkut Social Network," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining, pp. 678-684, 2005.

[21] Gang Qian, Shamik Sural, Yuelong Gu, Sakti Pramanik," Similarity between Euclidean and cosine angle distance for nearest neighbor queries".

[22] Anna Huang "Similarity Measures for Text Document Clustering" Department of Computer Science The University of Waikato Hamilton, NewZeland.

[23] Yuan Yan Tang , Bin Fang , Yong Xiang ,"Document Clustering in Correlation Similarity Measure Space" Knowledge and Data Engineering ,IEEE Transaction on (Volume:24, Issue:6)

[24] Mohammad Norouzi, David J. Fleet, Ruslan Salakhutdinov," Hamming Distance Metric Learning" Departments of Computer Science and Statistics University of Toronto

[25] Ankita Vimal, Satyanarayana R Valluri, Kamalakar Karlapalem , "An Experiment with Distance Measures for Clustering",Technical Report :IIT/TR/2008/132

[26] Rui Xu, Donald Wunsch "Survey of Clustering Algorithms" IEEE Transactions on Neural Networks , VOL. 16, NO. 3, MAY 2005.

[27] https://en.wikipedia.org/wiki/Jaccard_index.

[28] Gunes Erkan , Arzucan Ozgur, Dragomir R. Radev "Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing".

 All Rights Reserved © 2015 IJARCET

Ms. Syeda Nazema Syed Subhan received the BE degree in Computer Science and Engineering (CSE) from Dr. Seema Quadri Institute of Engineering and Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad in 2013. She is currently pursuing M.Tech in Computer Science, from Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Dr. Sachin N. Deshmukh is a assistant professor of Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad. He has completed his M.E. (Computer Science and Engineering) Ph.D. (Computer Science and Engineering). He is member of Professional Bodies: CSI, ACM, IETE, IAENG, and ACEEE. His research interests in Text Mining, Web Mining, and Social Network Data Mining.