Recommended System for Enhancing Tag Popularity in a Question Answering Community through Splay-net Techniques

Jayashree R

Abstract*: *Collaborative filtering filters information by using the recommendations of peer participants. The long tail problem states users with higher points obtain a high reputation compared to less scored users. In popular community question answering websites, like stack exchange network sites, users with unanswered or ignored questions for a long time get a tumbleweed badge without considering their past history. This deteriorates their further contribution to the website. Mostly new or low-reputation people ask the tumbleweed questions. The popularity of the tags follows a long tail theory. The focus of this research work is to design a recommendation system that prevents participants from tumbleweed badge with tag suggestion method to add new or non-popular tags to the existing popular tag list. The splay-net has a self-balancing graph which brings the recently accessed item to the top of the tree. In this paper, we use the splay-net technique to represent users' reputation along with their tags.

Keywords*: *Collaborative filtering; classification; Learning; Ranking system; splay Tree Data Structure.

I. INTRODUCTION

The basic idea of collaborative filtering is that the users agree on the evaluation of certain items in their future if they agreed on their past. In the neighbourhood-based technique, calculate the selected participant's average weight of the ratings to predict an active user. The reputation is a contextual measure of evaluating an individual's action which is a rough measurement of community trust on that individual. The reputation is based on the response of the community for one's deed which encourages one's good contribution to the community. Sometimes the reputation is considered as a measurement of knowledge, that is not always true. In some community-driven question and answering (CQA) websites, another use of reputation is to follow partaker's advancement on the site and reward them with added privileges. Since competitors are talented co-learners, gaining reputation is getting increasingly difficult. For some individuals reaching heights in reputation, a reputation cap is like winning a race. This perception is suitable for all grading scheme. In Stack Overflow website when a question (post) remains unattended for a longer time, then the owner of that question is rewarded with a batch called a tumbleweed, which is sometimes taken as a sign of disgrace. Mostly, less reputed partakers raise tumbleweed questions. After receiving this badge most of the partakers remain inactive on the website. In this paper, reputation award scaling is carried out to enticement the logjam. The goal is to apply the splay tree concept to add less used or new tags of the reputation cap lost partaker to the popular tag list and provide a tumbleweed prevention method. Compare to low-level systems based on dedicated machines Splay-Net provides accuracy, better scalability for distributed protocols, and ease of use [2], [16]. The authors in [3, 4] demonstrate that the low activity nodes have a low degree than high activity nodes through Code-based design (CB). The focus of CQA is on clustering which provides a platform to enable tagging the QA dataset [13]. There are numerous prevalent tags in the Stack Exchange network the analysis on which reveals the fact that tumbleweed partakers are mostly tagged with less prevalent tags. This fact is identified as one of the main factors for awarding tumbleweed badge. Only very few tags are popular in an active user group. The recommendation of popular tag in the tumbleweed prevention method will further increase the popularity of that set of tags alone without introducing new or unpopular tags to users. This results in a long-tail problem. The focus of the work is to introduce new tags in the popular tag list and to recommend a ranking system with the splay-net which is a graphical representation of the splay tree to prevent tumbleweed badge. We recommend a system for ranking with tumbleweed prevention methods. Select and include new or unpopular tags in the top tag list without affecting existing tags' popularity**.**

In the previous work, we presented an online rating calculation model for reputation management. This model approximates the expectation of the partaker's efficiency and provides simple update rules for ranking online using the Bayesian Approximation method and Normalized Discounted Cumulative Gain (NDCG) as a metric for measuring ranking correctness [25]. The excellence of the graded list is measured by Discounted Cumulative Gain which enables to forecast of the partakers' reputations. Sulthana and Ramasamy (2017) presented an algorithm to calculate the overall rating and grading of partakers. The NDCG method verifies this grading method.

The paper is systematized as follows: Section 2 explains the previous associated work. In Section 3 contributor splay-net with time complexity is described. Section 4 explains Tags popularity, and long-tail problem.

Revised Manuscript Received on December 05, 2020.
* Correspondence Author

Jayashree. R*, Department of Computer Applications, School of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur campus, Chennai, India. Email: jayashrr@srmist.edu.in.

The experiments and analysis of the results are described in section 5 and lastly, section 6 concludes and defines future work.

II. RELATED WORK

In splay tree leaf nodes corresponds to the symbol that is transmitted. These symbols move half-way towards the root which automatically pushes down the other symbols in the tree. The self-optimizing nature of the splay tree increases the performance notably (Dominic et al., 1985). Avin et al., (2013) described the concept of splay-net which is a distributed generalized splay tree with frequently communicated nodes are placed closer. The performance of Belief Measure of Expertise is evaluated on real data from Stack Overflow [15]. The only elements that categorize different topics in CQA are 'Tags'. Preslav et. al [6], [22] experiment on Question–Question similarity with the CQA Dup Stack dataset [23]. The concept of a self-regulating tree is almost best when the communication interests hold no conflicts and different communication patterns influence the concert of a distributed self- regulating tree [1]. The study on [7] shows that Segmented-Topic Model gives constantly improved concert related to the Latent-Dirichlet-Allocation Model for a dataset raised from the StackOverflow website. Choose the best answer to the question [24] and direct a novel question to probable finest specialists [7]. In Community Question Answering systems, the problem is routing new questions to the right group of experts. Fatemeh Riahi et. al (2012) overcomes this problem by presenting questions to the specialists matching their proficiency with the statistical topic models. Personalized summarization [5] proposes to amend the assessment history of a partaker based on the previous browsing history of a customer or by a partaker testimonial. Sulthana and Ramasamy (2017) find the evaluations connected to the favourite artifacts of the partaker and categorize their positive or negative ideas. Extracts the deeds of a partaker grounded on their fondness cart by constructing a cluster-tree for artifacts based on the partaker deeds in a customer based contextual- recommendation model [20]. As reported in the literature, more consistent and precise forecasts are the results of deploying a cluster of classifiers for choice making in a collaborative model [18], [19]. The excellence and extent of contributions of partakers are calculated by their reputations. A virtuous excellence contribution preserves the presented variations in subsequent revisions [10], [11]. Forecast upcoming contributions' excellence by estimating a partaker grade [10]. In the previous work, the prognostic ability measures the concert of the content- reputation model. Enhancing or reducing the influences of the large-degree partaker yields accurate reputation ranking lists [9]. Systematize the procedure of finding virtuous answers to novel questions in a Community question answering forum in [6]. The vector space model [21] represents the knowledge and similarity using the comparison of document vector and query vector.

Mostly the progression of the ICT-learning and its learning effectiveness assisted with the internet is the main reason for the people to gain awareness of the electronic-learning [25]. Not only the qualification but as the attitude of partakers improved through this electronic-learning. FUOLC's Electronic-learning provided a pathway for the development of software to the forecasted partaker's expectations through maintainability and reusability qualities [25]. They provided a context method that is based on Fuzzy logic to predict the instances and reviews that are relevant to each other with the association rules and ontology mining on textual analysis. The authors also compare the association between the review and the context using a semantic analyser based on the fuzzy rules. Over the data taxonomies, association rules like Apriori enable the review comparison on the distributed dataset [26]. Sulthana and Ramasamy (2017) presented a measuring metric for ranking correctness using Normalized Discounted Cumulative Gain (NDCG). The position of the recommended item in the overall list is important in the retrieval environment like software documentation application for personalization, and NDCG is often used along with item ranking place [17]. Reputation is defined as the total scores given by the peer partakers on the website and this reputation only ordered later to produce a ranking list [14]. Always authorized partakers to gain more mean points than unauthorized partakers since they are less trusted [14].

III. CONTRIBUTOR SPLAY-NET WITH TAGS

A tree with a splaying feature automatically shuffles the recently accessed nodes near to the root node. Represent participants with connected tags in a splay-net. Therefore nodes that are visited currently have BigO(1) as a reachable time when frequently searched. According to the locality-of-reference, 80% of the reachability depends on 20% of the node's location. Rotate the selected contributor 'c' with tags two levels high from her current position, so that her parent becomes her child. Though the rank changes, the c's reputation and thus, their trust values remain unchanged. Therefore, there are very few chances of receiving down-votes. In the previous work, we recommended an approach to encourage partakers of the community websites with a moderate position. When T is a tree with splaying feature and STi(p) is a node p's subtree at ith step, then p's rank at tree-level 'I' of splaying is

$$\mathrm{Rank}_i(p) = \log_2 (\mathrm{SizeOf}(\ ST_i(p)), \qquad (1)$$

where SizeOf(ST_i(p)) is the sum of weights of all elements in the subtree rooted in p. In other words,

$$2\mathrm{Rank}_i(p) = \mathrm{SizeOf}(\ ST_i(p))$$

$$p = \begin{cases} leaf, & \mathrm{Rank1(p)\ =\ 0} \\ root, & \mathrm{Rankn(p)\ =\ log2\ n} \end{cases} \qquad (2)$$

Where the sum of nodes in the tree is assumed as n_i. The ST's height and amortized-complexity for time are the factors that determine the time-complexity. The amortized complexity is based on the rank of the tree. Thus, the efficiency of a tree T, E_t(T), is the sum of the ranks of all its nodes.

$$E_t(T)=\sum_{i=0 \text{ to } n} \mathrm{Rank}_i \qquad (3)$$

The node depth d which is calculated from the node's position before and after applying the splaying concept,

$$d=|Pos_i \quad - \quad Pos_{(i-1)}| \qquad (4)$$

The Pos_i and $Pos_{(i-1)}$ are partakers' positions previous and later applying the splay-tree concept respectively. Let finding a 'key' acquires d time.

The node 'x' undergoes d/2 sub-steps while splaying. Alteration in the partaker move later applying the splay operation is $[Pos_i - Pos_{(i-1)}]$, calculate this only for the partaker, his parent, and his grandparent. Then the amortized-complexity is,

$$\begin{aligned} Amort_i &= e_i + (|Pos_i - Pos_{(i-1)}|) \\ &= e_i + d \quad \text{from (4)} \end{aligned} \tag{5}$$

Here e_i is an effort taken during the splaying, which is calculated from the number of levels the objective node ascends or descends during a splay operation. Total partaker move of a ST in jth step of splaying,

$$\begin{aligned} splay_i &= \sum_{p \epsilon ST_j} Rank_i(p) \\ &= \sum_{p \epsilon ST_j} \log_2 (SizeOf(ST_j(p)) \quad \text{from (1)} \end{aligned}$$

If step j initiates semi-splay operation then Amortized Complexity of splay tree is,

$$Amort_j < 1 + (Rank_j(p) – Rank_{j-1}(p))$$

In ST with m nodes, the amortized-cost Cost(m) of finding in a splay-tree is not more than $(1+ 3 \log_2 m)$ upward progress from the defined node. The Cost(m) is the number of replacement and Cost(m) = 1 if there is no replacement. At most (1 + 3d) is the Amortized-cost of a semi-splay if the number of replacements is one and the Amortized-time to splay a tree is at most,

$$3 (Rank(t) - Rank(x)) + 1 = BigO(log(SizeOf(ST_n(p)) / SizeOf(ST_x(p))))$$

where t is the root, x is a node and 1<=x<=d/2. Therefore, overall access-time AT and Amortized-complexity AC is,

$$\begin{aligned} AT &= BigO((n + e) \log e + n) \\ &= Cost(e) + E_t (ST) \quad \text{from (2) and (3)} \end{aligned}$$

That is the sum of amortized cost and potential difference, where e is the number of elements in the ST and n is the number of accesses.

$$\text{If } x, y > 0 \ \& \ z > x + y \rightarrow \log x + \log y <= 2 \log z – 2$$

$$AC <= \sum(Rank_i(x) – Rank_{i-1}(x)) + 1,$$

where 1<=i<=d/2.

For semi splaying,

$$3(Rank(t) – Rank(x)) +1 <= 3(Rank(t)) + 1 = 3 \log n + 1$$

Thus, if a1 is a splaying process and a2 is a semi-splaying process then the complexity of time is BigO((a1 + a2) log x). This is better than the conventional ranking model which provides BigO(ne) or BigO(e) as worst-case complexity.

Greater values of normalized discounted cumulative gain (NDCG) specify enhanced lists and that indicated enhanced perfection [17]. The discounted cumulative gain (DCG) studies the excellence of outcomes in a community-based graded list. Find out the system of measurements to signify the exact differences in performance between partakers. The cumulative gain is the summation of the scores for each partaker's place in the splay-tree. The discounted portion of NDCG adds the scores and divide it by the grade position. Frequently, the log of the rank divides the score, which appears to improve matching with the partaker's reputation. Modifications in the collation of search results do not disturb the value calculated with the Cumulative Gain (CG) function. Calculate DCG in the place of CG for a precise measure. The evidence of DCG is to penalize the ordering of partakers with a high grade at the lower level in the result of list-searching. Thus, the ranked reputation score lessens logarithmically proportional to the outcome location.

$$DCG_j = \sum_{j=1}^{n} \left(reputation_j - \frac{1}{\log (j+1)} \right)$$

In NDCG the normalized part compares multiple partaker's DCG values. The term ideal represents the best grading system which is otherwise called as $perfect\_DCG_k$. For a given set of m reputation, this is the most probable (perfect) DCG,

$$mDCG_j = DCG_j / perfect\_DCG_j$$

The position of partakers in the splay-tree helps in measuring the gain. Maximum DCG is the perfect_DCG (maximum possible). The NDCG value ranges from 0 to 1, with 1 representing the perfect ranking of the partakers. Select partakers along with their tags when their grade decreases from their previous value. Obviously, the value of NDCG< 0.5 for this ranking list.

IV. TAGS POPULARITY AND LONG-TAIL PROBLEM

StackOverflow distinguishes question categories using tags [12]. The overview of the literature [8] demonstrates how the best answer is selected in StackOverflow for the asked questions based on topic modeling and classifier. The focus of the work is on the expertise of the answerer by using TAG SCORE, the knowledge level metrics. In general, CQA award participants with the Tumbleweed badge without considering their old data. The two ways to prevent tumbleweed badge are the Tag Suggestion Method (TSM) and Edit or Delete Question Method (EDQM).

a. Tag Suggestion Method (TSM)

Research on tags and its popularity in community websites like StackExchange reveal the fact that partakers with a tumbleweed badge are frequently labelled with less popular tags. Thus, tags play an important role in the ranking system. The list of famous labels in community websites is extracted using the following query,

*select * from Famous_Labels*
where date >= date_from_parts(YEAR, MONTH,1) and label_name
in (select top T label_name from Famous_Labels
where datediff(month,date,getdate()) =1 order by count desc)
order by [date] asc;

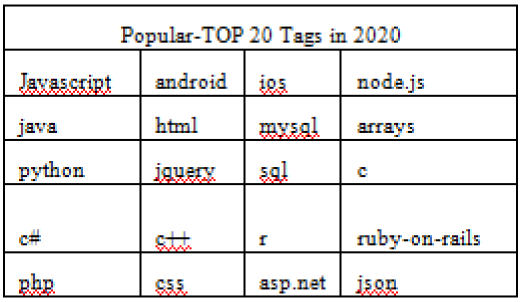
where T is any positive integer and *Famous_Labels* is a table with Date, Label_Name, and Label_Count attribute. Questions go unanswered when it is not labelled with the most famous tags. Therefore, advice about to receive tumbleweed or reputation cap lost users to include top tags. Table 1 and 2 lists out popular tags from the StackOverflow website in the year 2017 and 2020 respectively. Comparing both the tables shows that no new tags are added except a few changes in the order of tags. Tagging same topic in websites results in a long-tail problem.

Table 1. Top 20 tags in the year 2017

Popular-TOP 20 Tags in 2017			
Javascript	c#	mysql	node.js
Python	html	c++	Swift
Java	jquery	angular	Arrays

Android	css	sql	Json
Php	ios	r	python 3.x

Table 2. Top 20 tags in the year 2020



Popular-TOP 20 Tags in 2020			
Javascript	android	ios	node.js
java	html	mysql	arrays
python	jquery	sql	c
c#	c++	r	ruby-on-rails
php	css	asp.net	json

b. Edit or Delete Question Method (EDQM)

The percentage of unanswered questions is 73.96% and the community websites automatically delete zero score questions with no answers and a smaller number of views after a year of posting. Such a deleted question percentage is 7.8%. Most of the unanswered questions are deleted by the community website groups before closing them properly. Partakers are generally interested in answering recently posted questions under the familiar topic.

Thus, apparently, unanswered question edition and modification enable the questions to slip into the recently posted list and get prioritize at a different time of day, or on a different day which avoids tumbleweed. If the owner of the question is unable to edit then probably safe to delete it if it still has low views and no answers or up-votes.

V. EXPERIMENTS AND DISCUSSION

New contributors with new topics introduce new tags. Often about to receive or tumbleweed badge holders ask unique questions, mostly tagged with unfamiliar tags. According to TSM, using popular tags attracts the answerer. Most of the answerers are familiar with top tags topics. As a result, top tags gain more popularity and non-popular tags are not recognized by the community very easily. This long-tail problem is resolved by recommending non-popular tags to the top tag list through splaying. Intrigue contributor’s identity and their tags in a splay-net. Recommend preferred contributors’ top tags for including in popular tag list. If ‘n’ number of reputations lost users elect the same tag for ‘m’ number of times, then that tag weight is (n*m). The non-popular tag with the highest weight is nominated for the top tag list.

Table 3 lists sample tumbleweed awardees according to their grades Figure 1 and 2 shows their position among other ‘reputation-cap’ losers in a splay-net. In the Query-based system, the following query produces partakers arranged according to their reputation holding the Tumbleweed badge:

```
select top n p.id AS [Partaker Link], p.reputation
        from badges b
        join users p on b.partaker_id = p.id
       where b.Name = 'Tumbleweed'
       order by p.Reputation desc
```

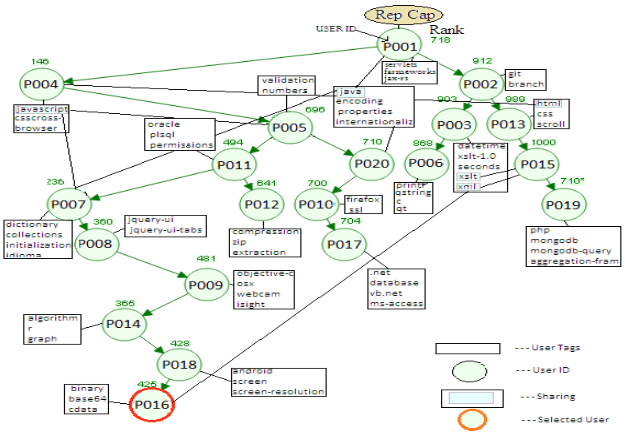


Fig. 1. Splay-net of contributors with sharing tag

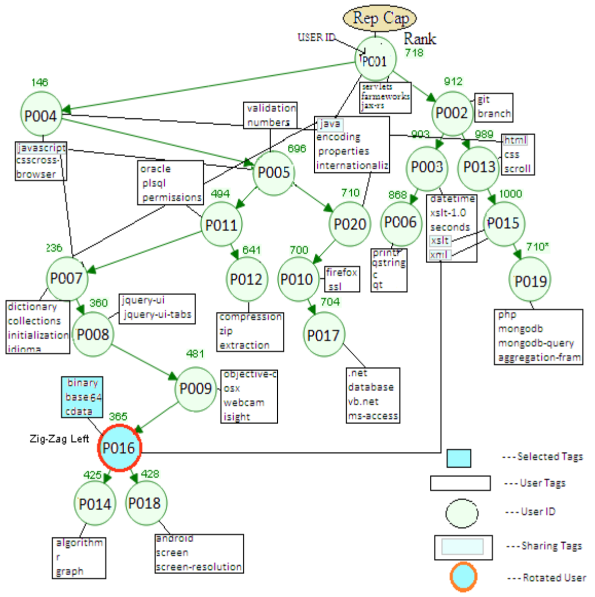


Fig. 2. Splay-net of contributors with sharing tags after rotation

Table 3. Top 20 user posts with 5 best tags

User ID	Posts	Rep	Percent Lost	% not-scored	Tags
P001	1747	81165	1	24.5	java, servlets, farmeworks, jax-rs
P002	2153	69739	10	9.7	git, branch
P003	3654	69995	13	33.2	xml, xslt, datetime, xslt-1.0, seconds
P004	1955	218108	6	7.2	javascript, html, csscross-browser, textselection
P005	3754	82985	18	24.5	javascript, validation, numbers

P006	574	72059	26	5.1	c, qt, printf, qstring
P007	1859	158960	16	7.6	java, dictionary, collections, initialization,idioma
P008	2989	121748	21	11.8	jquery-ui, jquery-ui-tabs
P009	2550	105796	34	9.3	objective-c, osx, webcam, isight
P010	1323	82421	50	9.7	firefox, ssl
P011	3481	103741	40	14.8	oracle, plsql, permissions
P012	2518	87055	49	11.1	java, compression, zip, extraction
P013	1305	66014	70	14.3	html, css, scroll
P014	1684	120784	40	6.9	algorithm, r, graph
P015	1366	65439	76	8.7	xml, xslt
P016	877	112779	51	1.5	xml, binary, base64, cdata
P017	2677	82210	75	13.1	.net, database, vb.net, ms-access
P018	3369	112577	59	15.6	android, screen, screen-resolution
P019	2139	74240	93	14.3	php, mongodb, mongodb-query, aggregation-fram
P020	714	81912	90	4	java, encoding, properties, internationalization

Calculate the reputation lost percentage from the sample stack exchange network metadata and compute uncapped reputation along with the percent-not-score percentage for Zero and non-zero accepted count for top 1000 reputation cap losers. Select and semi-splay a participant with minimum percent-not-score and with a good reputation when Normalized Discounted Cumulative Gain is less than 0.5 among n contributors, where n is 20. Semi-splaying the users' tags result in tag weight gain or loss accordingly. The goal of this paper is to identify the partakers who are eligible to take part in the reputation-cap using splay-net and to include the non-popular, blooming, tags in the popular tag list. This method periodically updates the tag list. This method also helps in the enhancement of partakers rank without affecting other peer partakers' reputation. Placed in the reputation cap is the sign of the procurement of trust. The contributor's splay-net is shown in Fig.2. It is built based on the minimum 'reputation lost' and 'sharing tags'. The 'sharing tags' are commonly shared by two or more users. Rotating any one of the sharing users modifies that tags' weight. The P001 is the minimum reputation lost partaker among all others thus concludes that this partaker is much closer to the reputation cap. Fig.2 shows the position of partakers in the splay-net after rotation along with selected tags.

Minimum reputation loss is not the only factor to consider, the previous history such as rank, reputation, and percent-not-score are also important factors for identifying P016 is potential than P001 to reach reputation-cap. Semi-splaying a contributor improves 6.2% of their probability to reach reputation-cap. The weight of selected users' tags increases. In our experiment, the tags are "xml, binary, base64, cdata". Since 'xml' is a 'sharing tag', recommend it for the top tag list.

VI. CONCLUSION AND FUTURE WORK

The Tumbleweed badge in stack exchange network websites is considered as a disgrace award by the website partakers. In this paper, splay-net of partakers and their tags provides a method to prevent or delay the issuing of this badge by the website. The primary focus of this paper is to study about relationship between Tumbleweed holders and their tags. Post of capable partakers are prioritized using splay-net concept along with a recommendation of their non-popular tag. The experimental result first proves that the semi-splaying method enhances the selected user's possibility to be in rep cap and recommends one of its top 'sharing tags' to get part of popular tag list. Tumbleweed prevention methods helps partakers who are about to get tumbleweed badge to great extent.

In future the work can be enhanced by selecting partakers based on their peer partakers' review along with context using fuzzy rules. A detailed study on tags with large amount of dataset is needed in future for better recommendation system.

REFERENCES

1. Thim Strothmann, "The Impact of Communication Patterns on Distributed Self-Adjusting Binary Search Trees", *Journal of Graph Algorithms and Applications*, vol. 20, no. 1, (2016), pp. 79-100, DOI: 10.7155/jgaa.00385
2. Valerio Schiavoni, Etienne Riviere, Pascal Felber, "SplayNet: Distributed User-Space Topology Emulation", *International Conference on Distributed Systems Platforms and Open Distributed Processing, Springer*, (2013), pp: 62-81.
3. Chen Avin, Alexandr Hercules, Andreas Loukas, Stefan Schmid, "Towards Communication-Aware Robust Topologies", *Research supported by the German-Israeli Foundation for Scientific Research and Development* (GIF), grant number I-1245-407.6/2014, (2017), pp. 1-14.
4. Chen Avin, Bernhard Haeupler, Zvi Lotker, Christian Scheideler and Stefan Schmid Ben Gurion, "Locally Self-Adjusting Tree Networks", *IEEE Computer Society*, (2013), pp. 395-406, doi: 10.1109/ipdps.2013.40.
5. Zouheir Trabelsi and Safaa Zeidan, "Multilevel Early Packet Filtering Technique based on Traffic Statistics and Splay Trees for Firewall Performance Improvement", *IEEE ICC - Communication and Information Systems Security Symposium*, (2012), 978-1-4577-2053-6/12 IEEE.
6. Preslav Nakov, oris Hoogeveen, Lluıs Marquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin and Karin Verspoor, "SemEval-2017 Task 3: Community Question Answering", Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), (2017), pages 27–48, Vancouver, Canada, Association for Computational Linguistics

7. Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei and Evangelos Milios, “Finding Expert Users in Community Question Answering”, WWW 2012 – CQA'12 Workshop, *The International World Wide Web Conference Committee* (IW3C2). ACM 978-1-4503-1230-1/12/04, (2012), pp. 791-798
8. Tirath Prasad Sahu, Naresh Kumar Nagwani and Shrish Verma, “Selecting Best Answer: An Empirical Analysis on Community Question Answering Sites”, *IEEE Access*, doi:10.1109/ACCESS.2016.2600622, Vol. 4, (2016), pp. 4797- 4808
9. X. L. Liu, *Ranking online quality and reputation via the user activity*, *Research Center of Complex Systems Science*, University of Shanghai for Science and Technology, Shanghai 200093, PR China, (2015), doi: 10.1016/j.physa.2015.05.043.
10. B. Adler and L. de Alfaro, “A content-driven reputation system for the Wikipedia”, Proceedings of the 16th *International World Wide Web Conference*, (WWW 2007), ACM Press, (2007).
11. G. Druck, G. Miklau and A. McCallum, “Learning to predict the quality of contributions to Wikipedia”, *Proceedings of AAAI*: 23rd Conference on Artificial Intelligence, (2008).
12. Hualei Dong, Jian Wang, Hongfei Lin, Bo Xu, Zhihao Yang, “Predicting Best Answerers for New Questions: An Approach Leveraging Distributed Representations of Words in Community Question Answering”, *Ninth International Conference on Frontier of Computer Science and Technology*, (2015), pp. 13-18, doi: 10.1109/FCST.2015.56
13. Alexander Beloborodov, Pavel Braslavskil and Marina Driker, “Towards Automatic Evaluation of Health-Related CQA Data”, CLEF 2014, LNCS 8685, *Springer International Publishing Switzerland*, (2014), pp. 7–18.
14. A. Razia Sulthana, and Subburaj Ramasamy, “Context Based Classification of Reviews Using Association Rule Mining, Fuzzy Logics and Ontology”, *Bulletin of Electrical Engineering and Informatics*, ISSN: 2302-9285, Vol. 6(3), (2017), pp. 250~255, doi: 10.11591/eei.v6i3.682.
15. Dorra Attiaouia, Arnaud Martinb, Boutheina Ben Yaghlane, “Belief Measure of Expertise for Experts Detection in Question Answering Communities: case study Stack Overflow”, *International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, KES2017, France, ScienceDirect - Procedia Computer Science, (2017), pp. 622–631, 10.1016/j.procs.2017.08.099
16. Daniel Dominic Sleator and Robert Endre Tarjan, “Self-Adjusting Binary Search Trees”, *Journal of the Association for Computing Machinery*, AT&T Bell Laboratories, ACM 0004-541 1/85/0700-0652, Vol.32(3), (1985), pp: 652-686.
17. Martin P. Robillard, Walid Maalej, Robert J. Walker, Thomas Zimmermann, “Recommendation Systems in Software Engineering”, *Springer-Verlag Berlin Heidelberg*, ISBN 978-3-642-45134-8 (eBook), (2013), doi: 10.1007/978-3-642-45135-5.
18. R. Polikar, “Ensemble Based Systems in Decision Making”, *IEEE Circuits and Systems Magazine*, Vol.6, (2006), pp. 21-45.
19. A. Quteishat, C.P. Lim, and J.M. Saleh, “A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement”, *Soft Computing*, Vol. 15(2), (2011), pp:221–231.
20. L. Song, C. Tekin, and M. van der Schaar, “Online Learning in Large-scale Contextual Recommender Systems”, *IEEE Transactions on Services Computing*, Vol. 9(3), (2016), pp: 433-45.
21. Jovita, Linda, Andrei Hartawan, Derwin Suhartono, “Using Vector Space Model in Question Answering System”, *Procedia Computer Science, Science Direct, International Conference on Computer Science and Computational Intelligence* (ICCSCI 2015), Vol. 59, (2015), pp. 305 – 311, doi: 10.1016/j.procs.2015.07.570.
22. Nakov, Preslav and Hoogeveen, Doris and Moschitti, Alessandro and Mubarak, Hamdy and Baldwin, Timothy and Verspoor, Karin, “Task 3: Community Question Answering”, *Proceedings of the 11th International Workshop on Semantic Evaluation*, Vancouver, Canada, (2017).
23. Doris Hoogeveen, Karin, M. Verspoor, and Timothy Baldwin, “CQADupStack: A benchmark data set for community question-answering research”, *In Proceedings of the 20th Australasian Document Computing Symposium*, Parramatta, NSW, Australia, ADCS ’15, Vol.3, (2015), pp 1–8.
24. M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen, “Probabilistic question recommendation for question answering communities”, *In Proceedings of the 18th International Conference on World Wide web*, WWW ’09, (2009), pp. 1229–1230.
25. Irma Salamah and M. Aris Ganiardi, “Development of Electronic learning Software Based Multiplatform Components”, *Bulletin of Electrical Engineering and Informatics*, ISSN: 2302-9285, Vol. 6(3), (2017), pp: 228~234, doi: 10.11591/eei.v6i3.647. 30
26. S. Kotsiantis, and D. Kanellopoulos, “Association Rules Mining: A Recent Overview”, *GESTS International Transactions on Computer Science and Engineering*, Vol. 32(1), (2006), pp: 71-82.

AUTHORS PROFILE

Dr. R. Jayashree, PhD, is an assistant professor at the department of Computer Applications, SRM Institute of Science and Humanities. She has served as coordinator for conducting various events in the National Level Symposium and Workshops organized by SRM IST. She published many papers in International and National Journals. She has participated and presented papers in International and National Conferences. Her area of interests includes Machine learning, Data Mining and Computer Graphics**.**