2017 International Workshop on Complex Systems and Networks (IWCSN)
December 8-10, 2017, Doha, Qatar

Predicting the Popularity of Tags in StackExchange QA Communities

Chenbo Fu, Yongli Zheng, Shidi Li and Qi Xuan
College of Information Engineering
Zhejiang University of Technology
Hangzhou, China 310023
Email: xuanqi@zjut.edu.cn

Zhongyuan Ruan
College of Computer
Zhejiang University of Technology
Hangzhou, China 310023
Email: zyruan@zjut.edu.cn

***Abstract*—StackExchange is one of the most popular Question and Answering (QA) websites, where each community address the questions on specific domain, e.g., programming, math, game, and so on. In these communities, users can use tags to label questions, which facilitates the search of questions and recommendation of experiments. Some tags are frequently used and thus get more and more popular with time, while some others are seldom used and finally diminish. The goal of this study is to find out the features that affect the future usage of tags and then design the popularity prediction algorithms. We investigate *structural* and *non-structural* features of tags, and using machine learning methods to classify popular and unpopular tags. The results show that, in general, the prediction models based on both structural and non-structural features indeed behaves better than those just based on one type of features, and the random forest (RF) method behaves the best among all the four considered machine learning methods.**

Keywords*—QA community; popularity prediction; machine learning; tag network; structural property.

I. Introduction

With the development of Internet and cloud technology, communities of people are virtually organized around and work on a common goal. Such communities include StackExchange [1], Open Source Software (OSS) projects [2]–[4], and so on, where people collaborate to provide quick and high-quality answers and create software, respectively, so as to form task-oriented social networks [5].

StackExchange is one of the most popular Question and Answering (QA) websites, containing a number of QA communities in different special domains. In these QA communities, tags (e.g., python, ios8, swift2) play an important role in filtering information. Typically, they are selected by the users to broadly cover the domains of questions, and a good tag will help the related questions to be easily searched and got the satisfied answers. Some tags are frequently used and thus get more and more popular with time, while some others are seldom used and finally diminish. An effective tag classifier will help the administrator to better manage the tags or design tag recommendation algorithms, e.g., Flickr tags [6] and Twitter hashtag [7].

Predicting the popularity is a well-defined problem in the area of social media [8], [9] and scientific community [10]. It is challenging since, in most dataset, only a very small portion of objects gain significant attention and thus get popular, while most of them gain little attention and thus diminish with time, making the two classes unbalanced. Most previous studies just utilized the *non-structural* features to design the prediction models, while in the area of network science, a series of *structural* features have been proposed, such as node centrality [11], clustering coefficient [12], assortativity [13], and so on. These structural features characterize the relationship between objects and thus may help to improve the performance of prediction algorithms.

Recently, Das *et al.* [9] proposed machine learning models based on the non-structural features measured over one day and then used them to predict the trends of hashtag on next day. Our tag popularity classification algorithms for StackExchange try to solve the similar task, but using both non-structural and structural features. The experiments validate that such algorithms performs better than those only based on non-structural or structural features.

The rest of paper is organized as follows. In Sec. II, we present the dataset and tag labeling. In Sec. III, we introduce all the non-structural and structural features that will be utilized to design machine learning algorithms. The experiments and results on tag popularity prediction are shown in Sec. IV. Finally, we conclude the paper in Sec. V.

II. Data Description and Tag Labeling

A. Datasets

Our study is based on the publicly available data from StackExchange. StackExchange provides data dumps of different QA communities. In each community, the data dump provides all posts, including information of questions and answers, tags, posting date, as well as information about user reputation and badges. Here, we mainly use the data dumps of the four largest QA communities, including Stack Overflow (SO), Ask Ubuntu (AU), Super User (SU) and Server Fault (SF). Their basic properties are presented in TABLE I.

B. Tag Labeling

In this paper, we mainly focus on utilizing both structural and non-structural features to establish classification model. We thus ignore those earliest emerged tags, since the tag

978-1-5386-1890-5/17/$31.00 ©2017 IEEE

TABLE I
BASIC PROPERTIES OF THE FOUR LARGEST QA COMMUNITIES IN STACKEXCHANGE.

Community	Description	Time Frame	# Tags	# Questions	# Answers	#Users	#Votes
SO	Stack Overflow is the largest QA community for programmers.	2008/07/31–2016/09/01	46,278	12,350,818	19,776,864	5,987,286	30,392,393
SU	Super User is a QA community for computer enthusiasts and power users.	2008/09/15–2016/09/01	5,197	325,245	486,237	455,052	3,024,484
SF	Server Fault is a QA community for system and network administrators.	2008/08/01–2016/09/01	3,447	227,375	388,944	277,093	1,922,178
AU	Ask Ubuntu is a QA community for Ubuntuusers and developers.	2009/01/08–2016/09/01	3,038	239,932	313,341	374,441	2,240,867

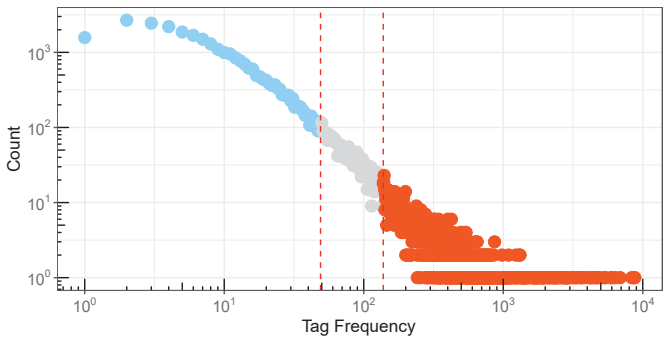

Fig. 1. The tag frequency distribution in SO community. The tags with frequency greater than 138 are chosen as popular tags (red) and those with frequency lower than 49 are grouped into an unpopular tag pool (blue).

networks at the times when they emerged were quite fragmentized, so that the local structure around these tags at those times cannot provide much information to predict their future evolution. Therefore, we only consider the 90% latest emerged tags. To label the popular and unpopular tags, we first sort all the considered tags by their usage frequencies within two years using descending order, and choose the top 5% as popular tags. Then, we put the 85% least frequently used tags into an unpopular tag pool. For example, in SO community, we choose the tags with frequency greater than 138 as popular tags and group those with frequency lower than 49 into an unpopular tag pool, as shown in Fig. 1. We can see that the tag frequency in SO community follows a power-law distribution. From the unpopular tag pool, we choose a corresponding unpopular tag closest to each popular tag in time, and thus get two balanced classes. Fig. 2 shows the word clouds of popular and unpopular tags in SO community.

III. TAG FEATURES AND PERFORMANCE METRICS

Generally, the most critical part of a machine learning model is feature selection. For the present work, we mainly focus on *non-structural* and *structural* features. The non-structural features are based on *natural attributes* of tags, and often used for best answers or experts detection [14]–[17]. On the other hand, the structural features are based on *network*

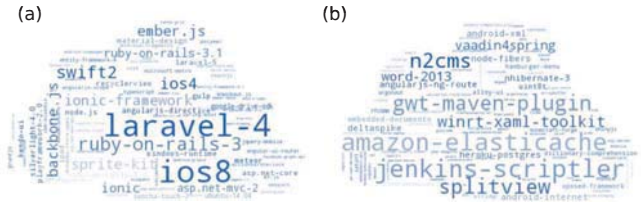

Fig. 2. Word clouds for (a) popular and (b) unpopular tags in SO community.

attributes, and are used for ranking important nodes or link prediction [18]–[20]. It should be noted that, our goal is to predict the future usage of tags based on the past records, thus all the features of a tag are extracted in a small time window τ since it was introduced for the first time.

A. Non-Structural Features

Non-structural features refer to the natural attributes of tags, including post and user features. Here, we consider the following features:

- **Number of Posts** (N_p)
  For each tag, the number of posts [15] is defined as the total number of questions and answers associated with it. Since the tag describes the topic of the post to certain extent, more posts associated with it means more popular of this topic.
- **Experience of Questioner** (E_q)
  For each tag, we first get all the questioners that have proposed questions using this tag. Then, the experience of questioner [16] is defined as the average number of tags they have used.
- **Number of Votes** (N_v)
  For each question q, we count the number of up votes as $N_v^{up}(q)$ and the number of down votes as $N_v^{down}(q)$, then the number of votes is defined as

$$N_v(q) = N_v^{up}(q) - N_v^{down}(q). \quad (1)$$

  Then, for each tag, the number of votes [21] is defined as the average number of votes of the questions using it.
- **Length of Question** (L_q)
  For each tag, length of question is defined as the average

number of words in the questions using it. This feature is also used to predict the popularity of tweets and online news [17], [22].

B. Structural Features

Before we extract the structural features, we first establish the tag network. In this study, we establish a quite simple tag network, where each node represents a tag and two nodes are connected if the corresponding tags belong to at least one same question. The link in this network thus has a weight, defined as the frequency of co-occurrence of two associated tags in the same questions. To describe the structural features of tags, we consider centrality and neighbor based attributes.

1) Centrality: Centrality is often used to identify important nodes in a network, e.g., finding important people in social networks, or key spreaders of epidemics [11].

- **Normalized Weighted Degree Centrality**
  The normalized weighted degree centrality of a tag t_i is defined as
  $$D_c(i) = \frac{\sum_{t_j \in N_e(i)} w_{i,j}}{N-1}, \tag{2}$$
  where $N_e(i)$ means the neighbor set of tag t_i; $w_{i,j}$ is the weight of the link between t_i and t_j; and N is the total number of tags in the network.
- **Eigenvector Centrality**
  Eigenvector centrality is also called eigencentrality, which captures the influence of a node in a network. In social networks, it means that the individual with many highly influential friends is also influential. The eigenvector centrality is defined as:
  $$E_c(i) = \frac{1}{\lambda} \sum_{t_j \in N_e(i)} a_{i,j} E_c(j), \tag{3}$$
  where λ is the maximum eigenvalue of the adjacency matrix $\mathbf{A}$ with the element $a_{i,j} = 1$ if t_i is linked to t_j, and $a_{i,j} = 0$ otherwise.
- **Closeness Centrality**
  It is defined as:
  $$C_c(i) = \frac{N-1}{\sum_{j \neq i} d_{i,j}}, \tag{4}$$
  where $d_{i,j}$ denotes the shortest path length between t_i and t_j in the network. The shorter the distances between tag t_i and the rest tags are, the more central the tag t_i is, and thus the larger this C_c index is.

2) Neighbor Based Properties: Neighbor based properties address the structural properties of neighbors of a node [12].

- **Cluster Coefficient**
  In social networks, cluster coefficient captures the dense of links between the neighbors of a node in a network, which is defined as:
  $$C(i) = \frac{2L_i}{k_i(k_i-1)}, \tag{5}$$
  where k_i is the degree of tag t_i, L_i is the number of links between k_i neighbors of t_i.
- **Average Clustering Coefficient of Neighbor**
  It is defined as:
  $$C_N(i) = \frac{\sum_{t_j \in N_e(i)} C(j)}{N_e(i)}, \tag{6}$$
  capturing the dense of links between the neighbors of neighbors of a tag.
- **Average Normalized Degree of Neighbor**
  It is defined as:
  $$D_N(i) = \frac{\sum_{t_j \in N_e(i)} D_c(j)}{N_e(i)}, \tag{7}$$
  capturing the importance of the neighbors of a tag.

C. Performance Metrics

We use two traditional metrics to measure the goodness of classification, i.e., *Accuracy* and *F1-Measure*. For binary classification problems, accuracy and F1-Measure can be computed by the four variables, TP (True Positive), FP (False Positive), TN (True Negative) and FN (False Negative) [23]. TP and FP refer to the numbers of Predicted Positives that are correct and incorrect, respectively. And it is similar for TN and FN. Accuracy represents the proportion of the correctly classified tags and can be defined as:

$$A = \frac{TP + TN}{TP + FP + FN + TN}. \tag{8}$$

F1-Measure is always used to measure the performance of a classifier since it is computed as the harmonic mean of *Precision* and *Recall* [24]. Precision P, Recall R, and F1-Measure are defined as:

$$P = \frac{TP}{TP + FP}, \tag{9}$$

$$R = \frac{TP}{TP + FN}, \tag{10}$$

$$F_1 = \frac{2PR}{P + R}. \tag{11}$$

IV. Method and Results

After extracted the *structural* and *non-structural* features of tags, we adopt four machine learning methods to design the popular tag classifier, including *Logistics Regression* (LR), *Support Vector Machines* (SVM) [25], [26], *Random Forest* (RF) [27] and *AdaBoost* (AB) [28], [29]. We test our methods on the four QA communities, i.e., SO, AU, SU and SF. It should be noted that our goal is to predict the popularity of new tags by utilizing their early behavioral patterns, therefore, for each new tag, we extracted its features in the first two months (Time window $\tau = 60$ days) since it was introduced. For each community, we randomly split the data into a training set and a test set, which contain 80% and 20% of the data, respectively. Then, we set the features mentioned in Sec. III as the input of the machine learning methods. By adopting LR, RF, AB and SVM, we establish the prediction models based on the training set and use these models to predict the popularity of the tags in the test set. For SVM in each community, we train

TABLE II
THE PREDICTION ACCURACY USING ONLY NON-STRUCTURAL OR STRUCTURAL FEATURES, FOR EACH METHOD ON EACH QA COMMUNITY.

Community	LR		RF		AB		SVM	
	Non-Structural	Structural	Non-Structural	Structural	Non-Structural	Structural	Non-Structural	Structural
SO	0.7358	0.6442	0.7313	0.7082	**0.7495**	0.7161	0.7387	0.7137
AU	**0.8080**	0.7676	0.7947	0.7544	0.7699	0.7288	0.7765	0.7420
SU	0.7814	0.7703	0.7850	0.7672	0.7623	0.7506	**0.7896**	0.7766
SF	0.7954	0.7981	0.7704	**0.8235**	0.7362	0.8000	0.7785	0.8131

TABLE III
THE F1-MEASURE USING ONLY NON-STRUCTURAL OR STRUCTURAL FEATURES, FOR EACH METHOD ON EACH QA COMMUNITY.

Community	LR		RF		AB		SVM	
	Non-Structural	Structural	Non-Structural	Structural	Non-Structural	Structural	Non-Structural	Structural
SO	0.7278	0.5977	0.7234	0.6922	**0.7347**	0.6961	0.7106	0.6822
AU	**0.8085**	0.7533	0.7842	0.7363	0.7634	0.7215	0.7494	0.7141
SU	**0.7837**	0.7703	0.7794	0.7626	0.7604	0.7528	0.7731	0.7652
SF	0.7973	0.7825	0.7622	**0.8133**	0.7343	0.7923	0.7739	0.7902

TABLE IV
THE ACCURACY AND F1-MEASURE USING BOTH NON-STRUCTURAL AND STRUCTURAL FEATURES, FOR EACH METHOD ON EACH QA COMMUNITY.

Community	LR		RF		AB		SVM	
	Accuracy	F1-Measure	Accuracy	F1-Measure	Accuracy	F1-Measure	Accuracy	F1-Measure
SO	0.7420	0.7339	**0.7614(1.59%)**	**0.7492(1.97%)**	0.7589	0.7527	0.7412	0.7113
AU	0.8143	**0.8101(0.20%)**	**0.8164(1.04%)**	0.8049	0.7938	0.7870	0.7783	0.7435
SU	0.7922	0.7891	**0.8132(3.00%)**	**0.8072(3.00%)**	0.7969	0.7950	0.8053	0.7955
SF	0.8163	0.8094	**0.8396(1.96%)**	**0.8307(2.14%)**	0.8146	0.8116	0.8165	0.7971

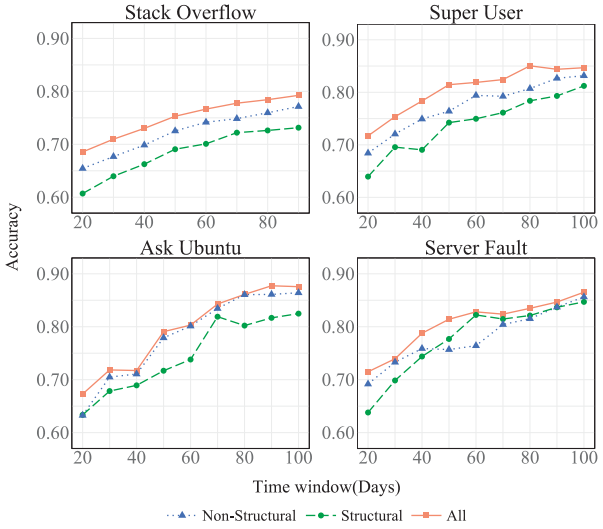

Fig. 3. The prediction accuracies as functions of time window τ by adopting RF, using only structural features, only non-structural features, and both of them, respectively, for the four QA communities.

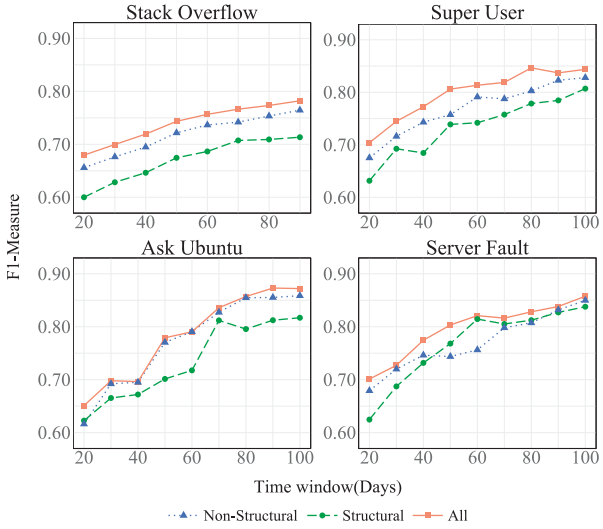

Fig. 4. The F1-Measures as functions of time window τ by adopting RF, using only structural features, only non-structural features, and both of them, respectively, for the four QA communities.

the model using 10-fold cross-validation to find the optimal parameters. All the models are generated by R packages (*stats*, *randomForest*, *adabag* and *kernlab*). Note that the parameter *ntree* of RF and the *kernel* of SVM are fixed and set to 1000 and *rbfdot*, respectively. Each experiment is repeated for 50 times and the mean values are recorded.

Since we have structural and non-structural features, we compare the methods by using only structural features, only non-structural features, and both of them. The results show that in SO, AU and SU, the performances obtained by the machine learning methods using non-structural features are slightly better than using structural features, while in SF, it

is better to use structural features, as shown in TABLE II and III. When using all the features, the prediction models behaves even better than those only based on structural or non-structural features, and comparing with the best results just based on one type of features, the best accuracies based on both types of features are improved by 1.59% (SO), 1.04% (AU), 3.00% (SU) and 1.96% (SF), and the best F1-Measures are improved by 1.97% (SO), 0.20% (AU), 3.00% (SU) and 2.14% (SF), as shown in TABLE IV. The results also indicate that the RF method always behaves better than the other three machine learning methods.

Furthermore, we also investigate the influence of the time window τ on the results. In the Fig. 3 and Fig. 4, we present the accuracy and F1-Measure as functions of time window τ by adopting RF. We find that the prediction models using both non-structural and structural features always perform better than the models just based on one type of features; and better prediction performances can be obtained when the time window is larger. This is reasonable since more information can be integrated as the time window gets larger.

V. Conclusion

In this paper, we adopted the four machine learning methods, including LR, RF, AB and SVM, by utilizing the non-structural and structural features to realize the tag popularity prediction in the four QA communities of StackExchange. We found that integrating both structural and non-structural features into the models can indeed improve the prediction results; and by comparison, RF behaves better than the other three machine learning methods. One limitation of this study is that the training and test sets are all from the same communities, therefore, we need to train a new prediction model for each QA community, which is not convenient in many applications. Fortunately, *transfer learning* has recently emerged to address this problem and get great success [30]. In future work, we will use transfer learning to further improve the practicability of our methods.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (11505153, 61572439, 11605154), Zhejiang Provincial Natural Science Foundation of China (LQ15A050002), and the Control Science and Engineering Discipline Prior Discipline of Zhejiang Province (20170706).

References

[1] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov, "How social q&a sites are changing knowledge sharing in open source software communities," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014, pp. 342–354.

[2] Q. Xuan and V. Filkov, "Building it together: Synchronous development in oss," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 222–233.

[3] Q. Xuan, A. Okano, P. Devanbu, and V. Filkov, "Focus-shifting patterns of oss developers and their congruence with call graphs," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2014, pp. 401–412.

[4] Q. Xuan, P. Devanbu, and V. Filkov, "Converging work-talk patterns in online task-oriented communities," *PloS one*, vol. 11, no. 5, p. e0154324, 2016.

[5] Q. Xuan, H. Fang, C. Fu, and V. Filkov, "Temporal motifs reveal collaboration patterns in online task-oriented networks," *Physical Review E*, vol. 91, no. 5, p. 052813, 2015.

[6] B. Sigurbjörnsson and R. Van Zwol, "Flickr tag recommendation based on collective knowledge," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 327–336.

[7] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings*, vol. 730, 2011, pp. 67–78.

[8] T. Trzcinski and P. Rokita, "Predicting popularity of online videos using support vector regression," *IEEE Transactions on Multimedia*, 2017.

[9] A. Das, M. Roy, S. Dutta, S. Ghosh, and A. K. Das, "Predicting trends in the twitter social network: A machine learning approach," in *International Conference on Swarm, Evolutionary, and Memetic Computing*. Springer, 2014, pp. 570–581.

[10] T. Chakraborty, S. Kumar, P. Goyal, N. Ganguly, and A. Mukherjee, "Towards a stratified learning approach to predict future citation counts," in *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*. IEEE Press, 2014, pp. 351–360.

[11] L. D. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in physics*, vol. 56, no. 1, pp. 167–242, 2007.

[12] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[13] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2006.

[14] T. P. Sahu, N. K. Nagwani, and S. Verma, "Topical authoritative answerer identification on q&a posts using supervised learning in cqa sites," in *Proceedings of the 9th Annual ACM India Conference*. ACM, 2016, pp. 129–132.

[15] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community qa," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 411–418.

[16] D. Correa and A. Sureka, "Fit or unfit: analysis and prediction of'closed questions' on stack overflow," in *Proceedings of the first ACM conference on Online social networks*. ACM, 2013, pp. 201–212.

[17] K. Fernandes, P. Vinagre, and P. Cortez, "A proactive intelligent decision support system for predicting the popularity of online news," in *Portuguese Conference on Artificial Intelligence*. Springer, 2015, pp. 535–546.

[18] Q. Xuan, C. Fu, and L. Yu, "Ranking developer candidates by social links," *Advances in Complex Systems*, vol. 17, no. 07n08, p. 1550005, 2014.

[19] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[20] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[21] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 621–630.

[22] H. Alharthi, D. Outioua, and O. Baysal, "Predicting questions' scores on stack overflow," in *CrowdSourcing in Software Engineering (CSI-SE), 2016 IEEE/ACM 3rd International Workshop on*. IEEE, 2016, pp. 1–7.

[23] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[24] F. Calefato, F. Lanubile, and N. Novielli, "Moving to stack overflow: Best-answer prediction in legacy developer forums," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2016, p. 13.

[25] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[27] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.
[29] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 161–168.
[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.