TAGGIT: Intelligent Interactive System for Stack overflow

Hariharan Balasubramani
Arizona State University
1213045871
hbalasu2@asu.edu

Srinidhi Sridharan
Arizona State University
1213054295
ssridh38@asu.edu

Srividhya Swaminathan
Arizona State University
1213134999
sswami10@asu.edu

Thiviyakalyani Navaneethan
Arizona State University
1213168188
tnavanee@asu.edu

ABSTRACT

Several question answering forums serve as good mediums wherein confused programmers can seek answers.Appropriate question structuring and categorizing under relevant tags is necessary to get fitting answers from such forums. Our system is an intelligent, interactive visualization that helps solve the purpose using two pivotal modules namely question quality prediction and tag comparison.

KEYWORDS

Stack Overflow, Tags, Question quality, Intelligent Visualization, Classification, Machine Learning models, Aesthetics, Interactive systems

1 INTRODUCTION

Stack overflow is a website that serves as a platform wherein users are allowed post their questions and seek answers for them. This also serves as an open medium where developers in the same community can collaborate to learn from other users'mistakes and doubts. For most efficient usage of this platform, a user should be aware of the correct methodology for posting a question such that it increases the likelihood of it being answered.

This is extremely vital for seeking an answer because a good quality question is the one that is aimed at clarifying some technical doubt (syntactical or logical). The question should trigger a debate - for example: “Which is better for front end development- angular or react?”. Such questions would make the users engage in aimless arguments than trying to debug or resolve a technical fault.

The other aspect of a good question is to classify it under the right tag. Some questions can fall under different categories or tags and hence it is important to choose the most appropriate tag for categorizing the question. For instance, a question in java spring might fall under “java”, “spring” and “mvc”.It is of primary concern to select the tag that has a very good developer community who would help the user get the right answer.

Our interactive intelligent system does this by offering two main functionalities. One is a tag comparator which the user can use to juxtapose the statistics of the selected tags and which in turn aids the user in categorizing his question under the appropriate tag. The second functionality being a question quality predictor, which predicts the question quality based on its meta data, which are used as features by the machine learning algorithm.

2 MOTIVATION

This project is aimed at helping a confused programmer who wants to post queries on a question-answering forum like Stack overflow. It is essential a programmer has a good understanding of how

a question should be structured. This can be inferred from the good quality questions which can be found using question quality prediction. A programmer needs to know the appropriate tag for their question in order to receive good results. In case a user has confusion between two tags, to solve his confusion there is need for a comparator that compares statistics for the two tags. The motivation is to solve the user's confusion and help them get better results from the forum.

3 METHODOLOGY

The Stack overflow dataset we are working on is composed of java questions. So, in order to filter out the tags, we choose the second tag in the tagset to be the tag of the question. For example, a tag that is “java android android-action bar” is taken to be under tag “android”. We choose the top 100 tags based on their frequency of occurrence. This is done because classifying questions in tags with too few questions will be difficult. The threshold for minimum number of questions is set at 300. The Tag Frequency is represented through an Interactive Bubble chart.

To know the appropriate structure of a question we have to first identify the good quality questions. The quality of the question is found using a intelligent computational model which in our case is a Machine Learning model. This model leverages the metadata of a question to predict its quality. For each question under a selected tag, the user should be able to view the results that were obtained. For this purpose, we have a bar chart which displays the answer acceptance rate for that answer. This also enables the user to compare the results for a good and bad quality questions in the past.

The Second functionality aims to compare two selected tags. This would help a user view the statistics of two tags side by side and make a choice of which would be a better option for them. If a user can know the trends in answers received for a tag, it helps them determine the quality of community for that tag. This be a deciding factor in their decision to choose a tag. A tag which has very large number of answers or shows a large positive trend in number of answers will be a better choice. This is done using a line chart. Knowing the composition of answer quality is also a factor in userâĂŹs choice. An answer with acceptance more than 50 is considered as a good answer. A tag with more good answers is likely to be a better choice. This composition of answers in a particular tag is done using a Donut chart.

4 IMPLEMENTATION

Data Analysis and Data Processing

The first step is gathering the data and transforming it into the most suitable form for processing. We have used the stack overflow dataset consisting of various questions, answers, accepted answers, the question length, code, text, tag, etc. The questions have to be grouped based on the tags and the number of questions under each tag was computed and the top 100 most frequent tags were considered for better results making it less cumbersome. The original file was comma separated consisting of the above-mentioned features and we transformed them to suit our application by computing length of question, each answers'up votes, etc. and discarding them for the less frequent tags.

Building the Computation Model

Making use of a model for the task of predicting the quality of a given question was vital and hence machine learning was used for this purpose. More specifically, we built a binary classifier that classifies the questions into two categories: good quality and bad quality. The metadata of the questions were considered as features for training the classifier. The accept_rate column of our processed dataset was used for assigning the good and the bad quality label. A threshold of 50 was set to demarcate the good quality ones from the bad quality ones. Three machine learning algorithms were considered for building this model namely Naive Bayes, Random forest and Support Vector Machine (SVM) and random forest yielded the maximum

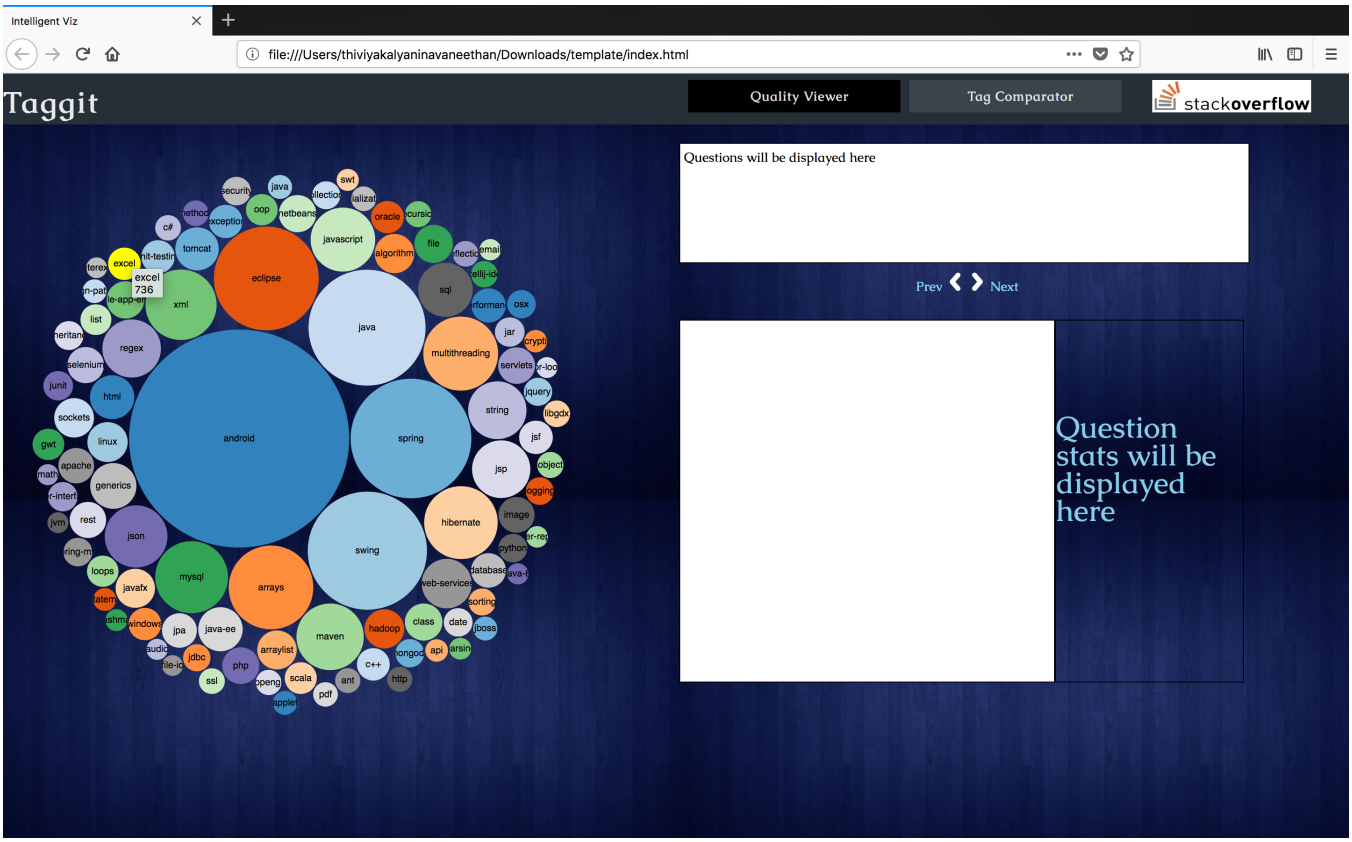

Figure 1: System

accuracy hence making it to the final model. The dataset was split into test and training set (80-20 split was used) and the test set is displayed interactively.

Visualization

Firstly, our system has two tabs namely - tag comparator and quality predictor. We display the top 100 tags as a word bubble and the sizes of the bubbles change according to its frequency. The bubbles can be viewed as filters and when they are clicked, the quality of the test set a question in the respective tags are displayed as a scrollable list. The user is allowed to click on a tag and the good and bad quality questions and further statistics is also presented by the means of a bar chart when hovered over those questions for gaining useful insights. In the second tab, when clicked on multiple tags, the numbers and facts about its metadata are displayed using donut charts and line chart. Line chart shows the number of answers under the chosen tags over a period of 12 months, while donut charts display the composition of the quality of answers.

Inference and Integration

The four charts have been integrated together as an interactive intelligent dashboard. The dashboard consists of two tabs each tab is used for a different functionality. The first tab is for a “tag comparator” has a bar chart wherein the bubbles are interactively used to select the tags and serve as a filter. The selected tags'statistics are displayed on the right-hand side with a donut chart showing the composition of good and bad quality questions.

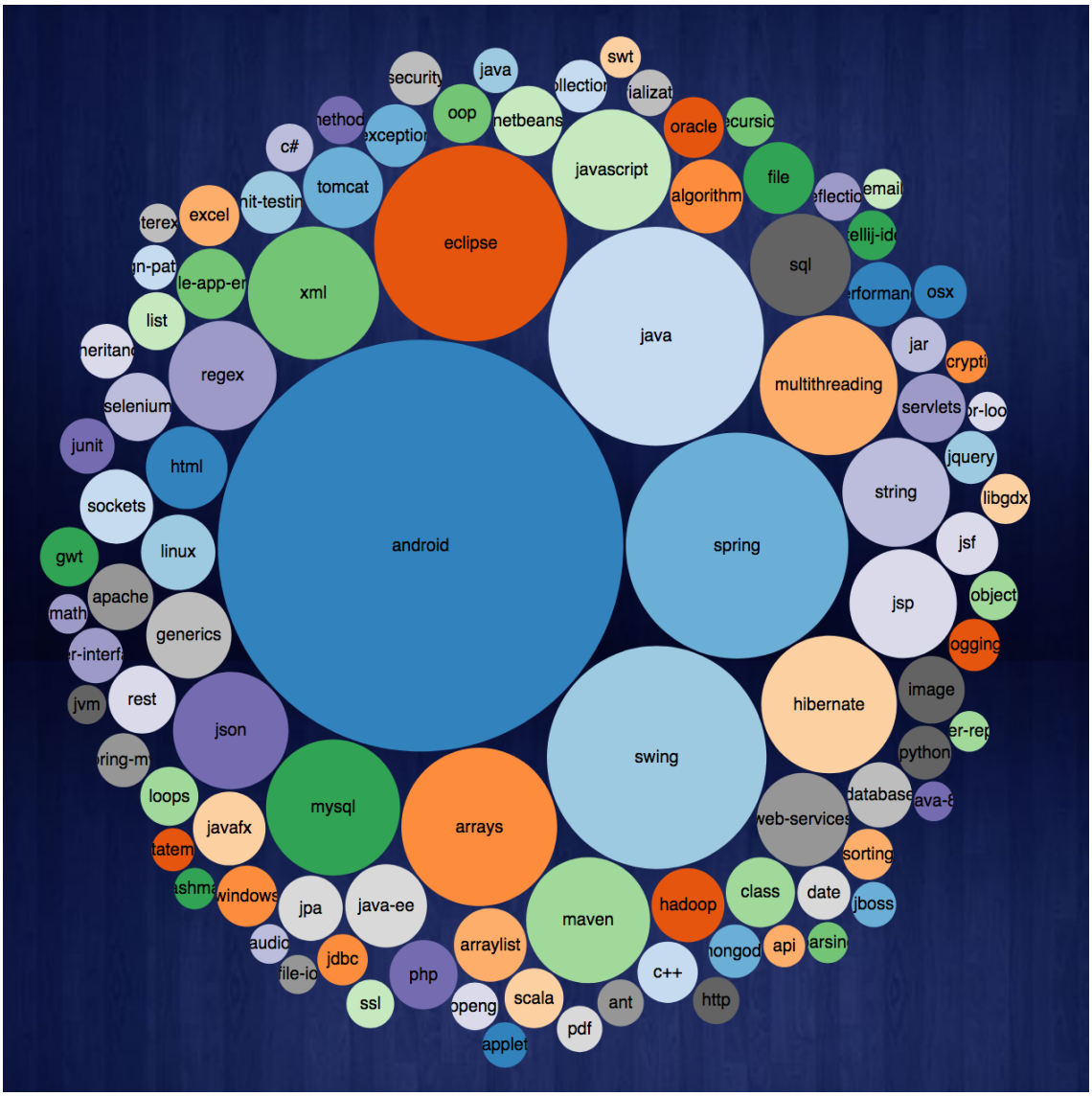

Figure 2: Tag Bubble Chart

A scatter plot is also used to show the evident correlation that exists. The second tab is a "question quality predictor", here the bubble chart acts as a filter and the selected tag's questions are displayed as a scrollable. The good quality questions are in green and the bad quality questions are in red colorable that is a bar chart displaying the stats of that particular tag.

The quality of a question can be predicted effectively by using its meta-data. We were able to do it with an accuracy of 80%. There exists a direct correlation between the quality of a question and the likelihood of the user getting an answer. An intelligent visualization system that does both prediction using a computational model and also visualizes it intuitively gives the maximum benefit for the user.

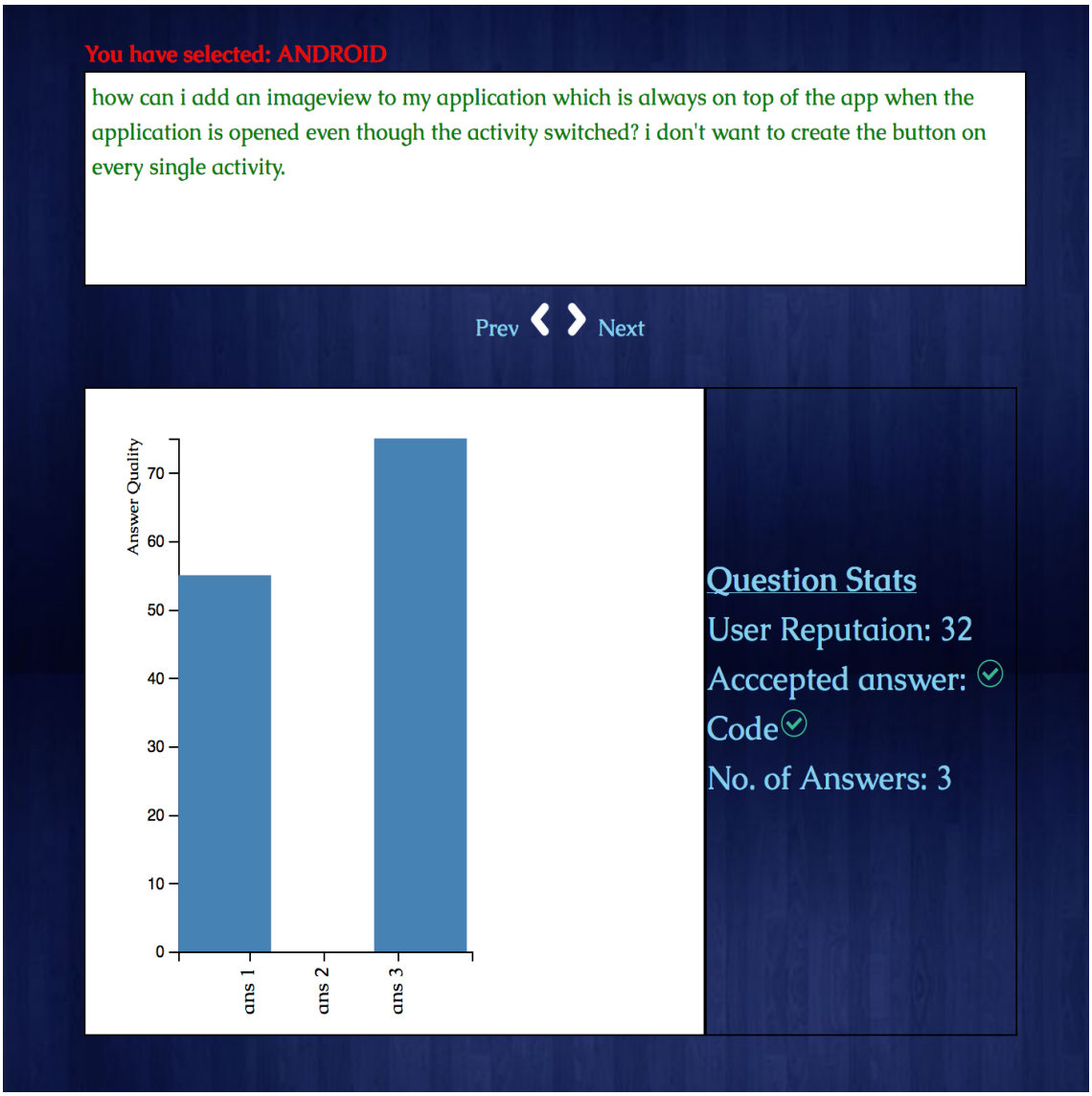

Figure 3: Question quality Scrollable and Answer quality Bar chart

5 EVALUATION

Building the computation model was attempted using 3 machine learning algorithms. For random 80-20 split dataset, we compared the accuracy of each model. Naive Bayes had 65% accuracy while SVM had 76% and Random Forest algorithm has 79% accuracy. Hence, Random Forest was chosen as the appropriate algorithm for this binary classification. Statistics for a question displayed along with the bar graph which represents the answer quality. These statistics are very helpful in getting the user to understand why that question was classified as good or bad quality one. The composition of answer quality in the donut chart is found to be an effective way of showing the quality of the community for that particular tag. More amount of good quality answers in a tag is indicative of more knowledgeable developers in the community. This in turn shows the user that there is more probability of getting a good answer for their question

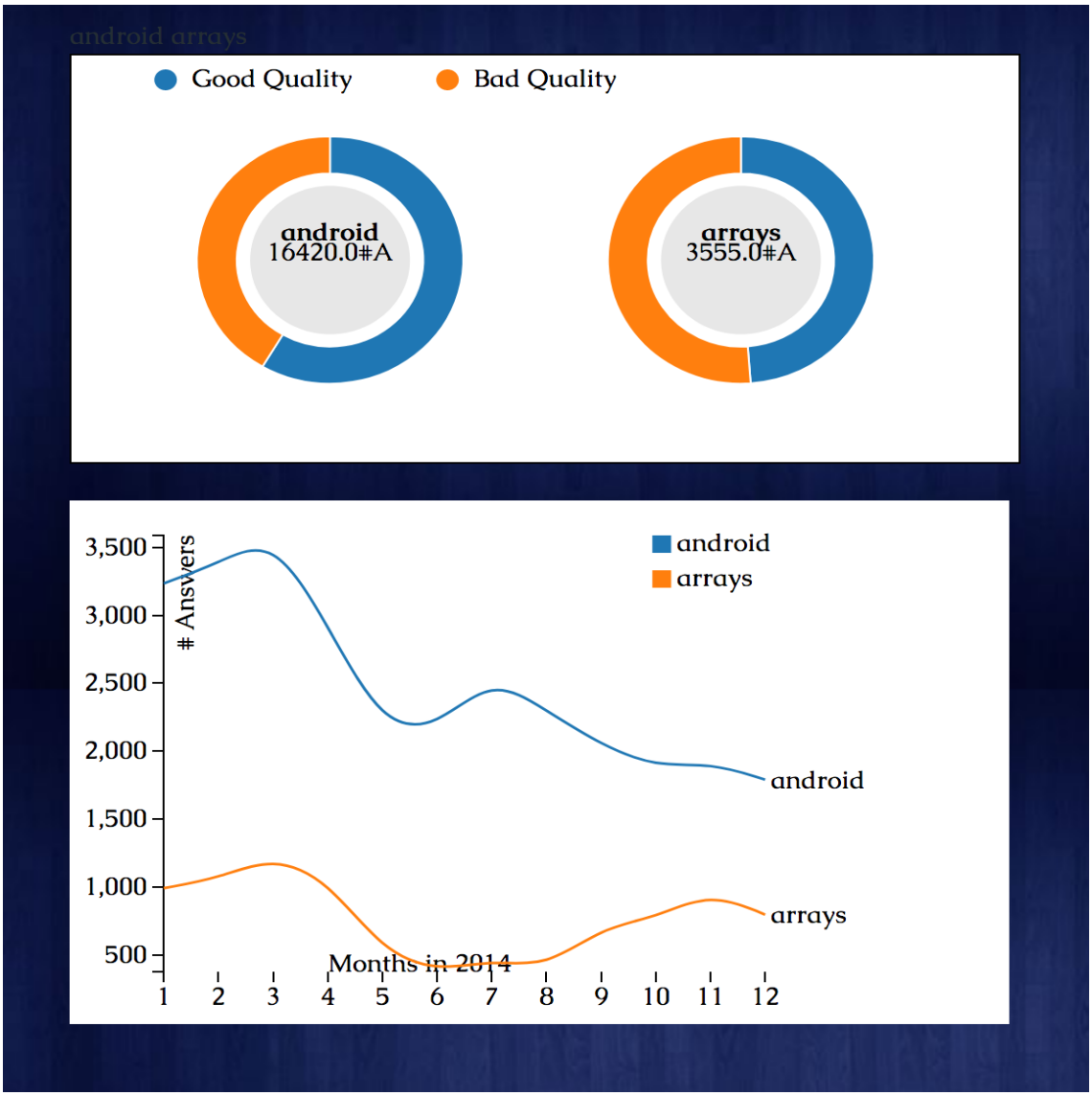

Figure 4: Tag Comparator

if put under this tag. The Line chart showing the trend of answers over the year shows us which time of the year sees more response from the community. For example, it is evident from the chart that there is a dip in the number of answers under a tag during the month of December. This trend is common for all the tags. It can be inferred that this general trend occurs due to December being the holiday season and hence, there would be a lesser probability of getting a good response during this month. The system is designed with colour schema that aesthetically please the user's eye.The Donut chart and Line chart use contrasting colours to represent the two different categories. This is done to make the difference visible easily. Similarly good and bad quality is colour coded as green and red which is representative of their quality. The colours chosen are bright colours in contrast to the background colour which is Dark Blue.

6 FUTURE WORK

The question quality predictor can be improved in future works with a model involving Natural Language Processing. NLP would build a model on the words and structures in the question and hence predict the question quality. This would also enable us to have a real-time model which can predict the quality of question the user enters and also give them feedback and possible improvements to the question. We can work on creating a more intuitive real-time system which given a user input question, shows questions that are similar to this that already exists.

REFERENCES

[1] Baltadzhieva et al, 2015; Baichuan Li et al, 2012; Sujith Ravi et al, 2014.

[2] Marek Lipczak et al, 2011; Short et al, 2014.

[3] Baltadzhieva, Antoaneta, and Grzegorz ChrupaÅĆa. "Predicting the quality of questions on stackoverflow." Proceedings of the International Conference Recent Advances in Natural Language Processing. 2015.

[4] Shah, Chirag, and Jefferey Pomerantz. "Evaluating and predicting answer quality in community QA." Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2010.

[5] Baltadzhieva, Antoaneta, and Grzegorz ChrupaÅĆa. "Question quality in community question answering forums: a survey." Acm Sigkdd Explorations Newsletter 17.1 (2015): 8-13.

[6] Ravi, Sujith, et al. "Great Question! Question Quality in Community QA." ICWSM 14 (2014): 426-435

[7] Stanley, Clayton, and Michael D. Byrne. "Predicting tags for stackoverflow posts." Proceedings of ICCM. Vol. 2013. 2013.

[8] Saha, Avigit K., Ripon K. Saha, and Kevin A. Schneider. "A discriminative model approach for suggesting tags automatically for stack overflow questions." Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on. IEEE, 2013.

[9] Barua, Anton, Stephen W. Thomas, and Ahmed E. Hassan. "What are developers talking about? an analysis of topics and trends in stack overflow." Empirical Software Engineering 19.3 (2014): 619-654.

[10] Anderson, Ashton, et al. "Discovering value from community activity on focused question answering sites: a case study of stack overflow." Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.