Analyzing StackOverflow Response Time for Java Topics using Code Clustering

Rezvan Ghaderi
University of California, Irvine
rghaderi@uci.edu

Shahrzad Ahmadpour
University of California, Irvine
sahmadpo@uci.edu

Aftab Hussain
University of California, Irvine
aftabh@uci.edu

ABSTRACT

We investigate relationships between StackOverflow question types and their response times. Since StackOverflow datasets do not provide detailed information about codes, except a tag that only determines the language of the code, we use the topic modelling approach to cluster codes in some groups. Then we perform statistical analyses to observe differences in response times for the clusters created by our algorithm. For the purposes of this study, we focus on a subset of questions related to Java which have been posted in 2013. Our findings will help to predict response times for new questions posted based on the cluster in which they fit.

General Terms

topic modelling, response time, anova, StackOverflow

1. INTRODUCTION

StackOverflow provides a convenient platform for software developers to post their programming questions and receive their answers from other developers and coding experts. Some of these questions are answered immediately, while there are lots of other questions which are not answered even after one day. Moreover, some questions are not answered at all. It is obvious that response time in StackOverflow and other similar Q&A websites has a very important role in user satisfaction. In this study, we are interested to research the factors which have effect on the response time of the posts in StackOverflow. There are many factors that might affect response time and some research has been done before for finding those factors. However, our study will focus on the code itself. Even questions which are related to one language such as java are very diverse in their scopes. Some of them might be related to simple question about using operators, while some others might be about working on files, using databases or network programming. Since StackOverflow dataset does not provide any detailed information about codes, except a tag that only determines the language of the code, we use "Topic Modeling" to cluster codes in some groups. Since topic modeling is based on word frequencies in similar documents for finding the abstract topics, we believe that this unsupervised algorithm will be a good choice for our purpose. The mean response times for each clusters were than compared using a statistical procedure, ANOVA, where we deployed null hypothesis testing. Our hypotheses and consequent outcomes are detailed in subsections 3.3 and 4.2 respectively.

This paper is organized as follows: in Section 2, we outline related work in this area. In Section 3, we present our research methodology and define the central concepts of our research. We give our results and conclusions in sections 4 and 5 respectively.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
INF 211 Software Engineering Project Fall 2014
Copyright 2014 ACM X-XXXXX-XX-X/XX/XX ...$15.00.

2. BACKGROUND AND MOTIVATION

Response time analysis on Q&A websites can assist site administrators in taking measures to speed up questions being answered. It can enable them to direct questions to those who possess the required expertise and thus expedite answering [1]. However, prediction of response time in a Q&A site is a challenging problem as there may be many contributing factors. Consequently studies on response times of questions asked on-line have been carried out on several fronts, some of which are briefly discussed as follows: (1) There has been studies on how response times varied with the domain-expertise of the answerer. For example, in [3] response times for questions asked to targeted users were analyzed. Users were targeted based on their domain expertise, which was determined based on feeds from their Twitter posts. Then, questions were sent to them from anonymous people. The study found 44% of these responses arrived within 30 minutes. (2) There has been studies where the relation between asker satisfaction and response time was investigated. E.g. Rechavi et al. [4], based on their analysis on Yahoo! Answers, found that askers are generally more satisfied with quick responses. (3) There were studies which analyzed the relation between response times and site-specific peripheral information explicitly provided with the question by the asker. E.g. Vasu et al. [1], found a relation between the tags provided by the asker and the response time of the question.

Our work closely relates to the third category of response time research, with the important distinction of focusing on the relation of response time with information that is *implicitly* provided with the question. In particular, we automatically identify the topic a question belongs to and find the average response time for all questions in that topic. For

this end we used a statistical topic modeling technique, the LDA, which is discussed in Section 3. Explicitly provided peripheral information in StackOverflow, like tags, are generally insufficient to get an understanding of the domain of the question as they may be too abstract or too detailed. Relying on a topic modeling technique will smoothen this constraint in topic identification, and thus facilitate more meaningful response time analyses. Although we focused on codes of 2013 StackOverflow Java Questions for this study, we believe our approach can be applied to a broader scope within StackOverflow and other Q&A websites as well. It should be noted that topic modeling techniques have been used in StackOverflow, but not for response time analysis. One example is [6] which analyzes similarities between user questions and answers in StackOverflow based on topic modeling techniques.

3. APPROACH

In this section, we present our research methodology which consists of three components: Data Extraction, Topic Modeling, and Analysis, which are explained in the following subsections.

3.1 Data Extraction

The data was extracted from the StackOverflow database available in the StackExchange online website [1]. All necessary queries were performed on the "Posts" table in order to extract 2013 question post bodies and corresponding response times. Then, by executing scripts, the post bodies were cleaned until they were left only with codes, which is the input of the next phase of our work. A total of 88,355 code snippets were processed for topic modeling.

3.2 Topic Modeling

We used topic modeling for the codes, without removing the comments in them. We created a stop word list of Java keywords that are common and could seen in any Java code snippet. These words were removed as they would not assist in delineating the snippet topics. For topic modeling we used the Latent Dirichlet Allocation (LDA) Technique.

Brief description of LDA. The LDA is a generative probabilistic model for collections of discrete, unobserved data such as text corpora. It can show some parts of this unobserved data are similar. The LDA relies on a three-level hierarchical Bayesian model where each item of a collection is modeled as a finite mixture over an underlying set of topics [2] and each topic is modeled as an infinite mixture over an underlying set of topic probabilities.

We ran LDA multiple times with different settings for number of clusters and alpha parameter (i.e. the parameter of the Dirichlet prior on the per-document topic distributions). Topics created in every step helped us to find some other words that which was general, therefore we added them to the stop words too. We selected the output of LDA with 15 number of clusters. This decision was based on the manual inspection of topics. In this step we selected the topics which better categorized codes into clusters, from a typical Java developer's perspective, based on popular Java library terms and variable names which described any category. The categories obtained are shown in Section 4.

[1]http://data.stackexchange.com/stackoverflow/query/new accessed in November 2014.

The Python library for textual topic modeling, Gensim, was used to implement the LDA technique on the codes. Gensim facilitates unsupervised semantic modeling from plain text. More information could be found in [5].

3.3 Analysis

We analyzed the response times for each topic cluster using analysis of variance (ANOVA). ANOVA is a statistical technique included in many introductory statistics courses, which analyzes the relationship between a quantitative dependent variable and one or more independent qualitative variables. The nature of the relationship is expressed in a model with unknown parameters [7]. ANOVA can be considered as an extension of the t-test to situations in which there are more that two groups to evaluate or there is more than one independent variable. The conceptual model for ANOVA follows the following pattern: a ratio is formed between the differences in the means of the groups and the error variance. In the same manner that a variance can be calculated form a set of data, a variance can also be calculated from a set of means.

In our experiment, we performed the ANOVA test using R, to compare the means of response times among the different groups. In fact, we want to know are the variations between different groups means due to the true differences about the response time means or just due to the sampling variability.

The following hypotheses were adopted for ANOVA:

H_0: *The means of all groups that we are experimenting is equal.*

H_a: *The means are not equal.*

We can figure either of them by analyzing the p value from the ANOVA test. ANOVA calculates a parameter called F statistics, which compares the variations between sample means among different groups to the variation within groups themselves.

4. RESULTS

In this section, we present our results of the data and elaborate upon the analysis techniques that have been used.

4.1 Clusters Obtained

Figure 4.1 shows the clusters that were obtained after deploying our topic modeling technique.

4.2 Statistical Analyses

The average response time, variance, and length (i.e. size) for each cluster is presented in Figure 4.2.

Among all the different clusters in our experiment, Text cluster has the minimum average response time and Servlet has the maximum average response time. Probably, StackOverflow users have a vast knowledge of Text and are able to respond quickly with 34.7 average response time compare to Servlet and project building with 213.3 and 182.2 average response time, respectively. Text cluster had the minimum standard deviation among all and Servlet had the maximum standard deviation among all the clusters. The difference between the maximum and minimum standard deviation was almost 576.

Android and graphics have very close average response time (68.65 and 67.79 hours respectively). Also HTML response time (109.23 hours) and JDBC response time (105.21 hours) were very close to each other. This shows that users

Figure 1: Clusters derived from Java codes.

Cluster 1: Android android, id, view, androidruntime, textview, app, activity, findviewbyid, button, activitythread	Cluster 2: HTML value, text, type, html, id, form, div, request, title, action	Cluster 3: Lists list, file, arraylist, get, map, add, put, size, hashmap, util	Cluster 4: Text length, scanner, input, number, array, count, temp, index, nextint, next	Cluster 5: Building error, info, eclipse, debug, build, users, target, path, src, home
Cluster 6: Tree object, data, item, row, node, table, parent, root, col, model	Cluster 7: Date date, time, calendar, format, mm, gson, getinstance, state, year, get	Cluster 8: Spring springframework, http, name, property, beans, artifactid, web, groupid, hibernate	Cluster 9: Servlet apache, sun, core, invoke, http, util, catalina, service, javax, servlet	Cluster 10: Json log, context, json, getstring, intent, toast, show, result, super, cursor
Cluster 11: XML name, id, type, column, element, xml, entity, doc, user, description	Cluster 12: GUI add, javax, swing, awt, jbutton, jpanel, frame, jlabel, jframe, event	Cluster 13: JDBC user, password, jdbc, username, select, sql, connection, query, result, driver	Cluster 14: Graphics image, float, player, height, width, math, color, graphics, game, size	Cluster 15: Thread lang, source, run, method, thread, net, google, exception, init, awt

Figure 2: Average, Number of Snippets, and Standard deviations for different clusters.

Cluster	No. of Snippets	Average Response Time (Hours)	Standard Deviation in Response Time
Android	5166	68.65303	529.4861
HTML	5323	109.23830	693.1440
Lists	8782	59.65556	608.9620
Text	8218	34.79418	440.7709
Building	5136	182.22836	914.1830
Tree	5588	73.43700	600.5637
Date	5847	68.17798	595.1666
Spring	4254	169.390	875.95
Servlet	2229	213.30050	1016.1048
Json	3586	92.31054	655.1145
XML	13354	84.03453	622.9344
GUI	6040	46.32701	490.6374
JDBC	5835	105.21018	797.5587
Graphics	5078	67.79722	529.7746
Thread	3918	132.51543	780.1744

with the knowledge of these fields have the same amount of activity in Stack overflow and can reply the posts in an approximately same time intervals. In addition, the standard deviations for Android and Graphics were almost the same (529.48 and 529.77 respectively).

Among all the clusters, XML had the biggest length with a significant difference compared to other clusters and Json cluster had the smallest length among all the clusters.

Figure 4.2 shows the plot for the average response time in terms of hours for each cluster. Figure 4.2 shows a box plot of the mean differences between cluster pairs.

So far we have recorded the average of response times. But the question is whether or not there is a significant difference in response time among them? To answer this question, we dive into the concept of Analysis of Variance (ANOVA).

First, we found the F statistic. Through the F statistic we can see if the variation among sample means dominates over the variation within groups, or not. It is given as follows,

$$F \text{ statistic} = \frac{\text{Variation among sample means}}{\text{Variation within groups}}$$

Figure 3: Average of response time for different clusters.

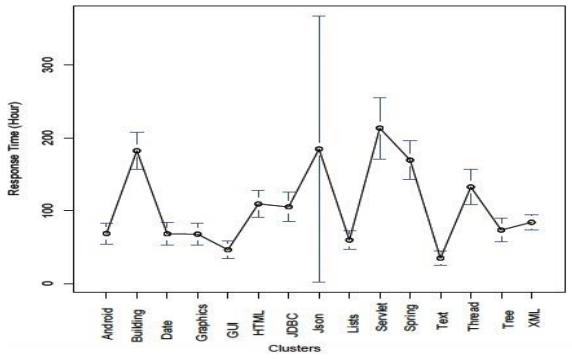
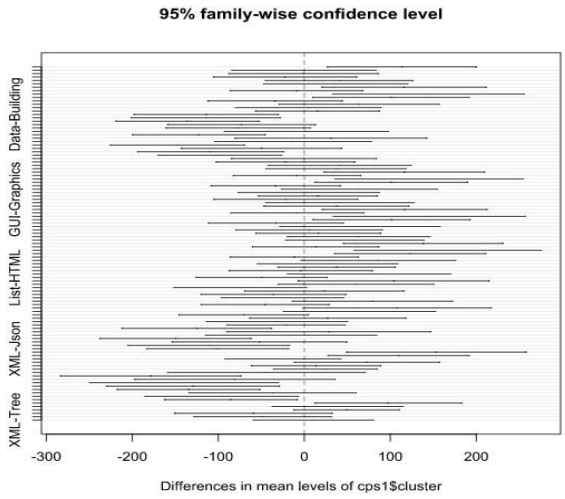


Figure 4: Differences in mean levels of clusters.



The F value was found to be 27.72, and p-value was very low too. In other words, the variation of response time means among different clusters (numerator) is much larger than the variation of response time means within each cluster, and our p-value is less than 0.05 as suggested by normal scientific standard.

ANOVA test results showed that 31 out of 105 pair comparisons had significant difference in response time with the p=0, which shows that the compared pairs need significantly different knowledge and the distribution of these knowledge is different among StackOverflow users. Also, this significant difference might come form the fact that how popular the area is. More people are working in more popular areas and the chance of the questions to be answered are getting better once more users work in the field. In addition, 7 out of 105 pair comparisons had no significant difference with p=1. Android, Tree, Date and Graphics clusters have been mostly seen among the compared pairs that resulted in no significant difference with p=1. Finally, we also visualized cluster pairs and analyze significant differences by plotting the "tuk" object in *R*. Significant differences are among the pairs which do not cross the zero value, and pairs which cross the zero value are not significantly different.

5. CONCLUSION

In this work we have analyzed relationships between StackOverflow question types and their response times. From 88,355 snippets extracted from all StackOverflow questions from the year 2013, in the area of Java, 15 question groups or clusters were determined. The LDA topic modelling approach was deployed on the codes in order to form the groups. Differences between the response times were assessed using standard statistical procedures. The study found some interesting findings, e.g. response times of Text related questions are the shortest, whereas response times of Servlet related questions are the longest. Our findings will help to predict response times for new questions posted based on the cluster in which they fit. Given that prediction of response time in a Q&A site is a challenging problem, our study takes a step in mitigating this challenge.

6. REFERENCES

[1] V. Bhat, A. Gokhale, R. Jadhav, J. Pudipeddi, and L. Akoglu. Min(e)d your tags: Analysis of question response time in stackoverflow. In *International Conference on Advances in Social Networks Analysis and Mining 2014*, pages 328–335, 2014.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, (3):993–1022, 2003.

[3] J. Nichols and J.-H. Kang. Asking questions of targeted strangers on social networks. In *In Proc CSCW 2012*, pages 999–1002, 2012.

[4] A. Rechavi and S. Rafaeli. Not all is gold that glitters: Response time and satisfaction rates in yahoo! answers. In *In SocialCom/PASSAT, IEEE*, pages 904–909, 2011.

[5] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[6] F. Riahi, Z. Zolaktaf, M. Shafiei, and E. Milios. Finding expert users in community question answering. In *CQA'12 Workshop 2012*, pages 328–335, 2012.

[7] R. Sturm-Beiss. A visualization tool for one- and two-way analysis of variance. *Journal of Statistics Education*, 13(1), 2005.