An Exploratory Analysis of Mobile Development Issues using Stack Overflow

Mario Linares-Vásquez, Bogdan Dit, Denys Poshyvanyk
Computer Science Department
The College of William and Mary
Williamsburg, VA, USA
{mlinarev, bdit, denys}@cs.wm.edu

Abstract—Question & answer (Q&A) websites, such as Stack Overflow (SO), are widely used by developers to find and provide answers to technical issues and concerns in software development. Mobile development is not an exception to the rule. In the latest SO dump, more than 400K questions were labeled with tags related to mobile technologies. Although, previous works have analyzed the main topics and trends in SO threads, there are no studies devoted specifically to mobile development. In this paper we used topic modeling techniques to extract hot-topics from mobile-development related questions. Our findings suggest that most of the questions include topics related to general questions and compatibility issues, and the most specific topics, such as crash reports and database connection, are present in a reduced set of questions.

Index Terms—Stack Overflow, mobile platforms, mining software repositories, topic modeling.

I. INTRODUCTION

Nowadays, mobile development is related to (i) a reduced set of technologies that allow developers to create mobile applications (apps) for specific software and hardware platforms, as well as (ii) a set of cross-platform tools that allow developers to use models or high-level programming languages to create multi-platform apps. Moreover, the different options of handset vendors that are available in the market have promoted the usage of several programming languages for mobile development (e.g., Android, iOS).

In the case of traditional software development, there are several paradigms, and consequently, different issues and concerns that are specific to the paradigm, the programming language, and the type of application to build. However, there is less evidence and knowledge related to the possible issues associated with mobile development than with traditional software development. In particular, we do not have solid evidence of the most common issues that developers face when developing apps for specific software platforms, such as Android, iPhone, or JavaME.

A recent study by Han et al. [1] analyzed fragmentation within Android by extracting topics from bug reports and using topic modeling techniques. Although the authors provide evidence of issues that affect the bugs reported in the Android bug repository, those issues are related to hardware-fragmentation and not include the software-fragmentation perspective. Barua et al. [2] used topic modeling to automatically extract the main discussion topics that software developers keep in Stack Overflow1 (SO). Stack Overflow is a question & answer (Q&A) website widely used by developers to find and provide answers to technical issues and concerns in software development. In the study by Barua et al. [2], a total of 973,267 questions and 2,501,720 answers, which represent 27 months of SO activity (from July 2008 to September 2010), were analyzed. The main topics identified by Barua et al. [2] in SO are related to: web-related discussions, data management, platform-specific discussions, security, quality assurance and collaboration, knowledge/experience, and general discussions. In addition, two of the conclusions drawn from the analysis were: (i) mobile application development is a trend topic, with an upward-tendency that increments faster than web development; (ii) Android and iPhone development is much more prevalent than Blackberry development. However, Barua et al. [2] did not explore the details related to mobile development found in the discussions of SO, and to the best of our knowledge, these details were not addressed in any other research paper.

In this paper, we plan to further explore the issues that developers face when developing apps, by extracting topics representative of issues in mobile development. More specifically, in this paper we analyzed the mobile-development-related discussions from SO, by extracting the main topics that represent those discussions using Latent Dirichlet Allocation (LDA) [3]. Our work is similar to Barua et al.’s [2] because it uses LDA, but is different from the type of information analyzed. Because SO is widely used for finding answers to technical issues in programming, our assumption is that the terms found in questions and answers describe the latent structure of the technical issues associated with software development for mobile devices.

Several tags have been used by SO users to label questions related to mobile development. In the latest SO dump (August 2012) there were more than 400K questions labeled with different tags related to mobile technologies (e.g., languages and cross-platform tools). We used those questions and their accepted answers to extract the main discussion topics at two granularity levels: for the entire dataset, and tag specific

1 http://stackoverflow.com/
corpora. We distinguished between questions that have accepted answers and questions without accepted answers. In addition to the topics analysis, we analyzed if SO contributors are more concerned to provide answers in specific technologies, or for multiple platforms.

All the data used in our study are publicly available at http://www.cs.wm.edu/sem eru/data/msr13-so-mobile.

II. DATA AND APPROACH

Our study aims at answering the following three research questions (RQ):

- **RQ1**: Are there developers that provide accepted answers for several mobile platforms? Do developers provide answers only for a specific platform? Our interest with this RQ is to validate if SO contributors use two or more platforms concurrently.

- **RQ2**: Which are the hot-topics that describe the answered questions related to mobile development in Stack Overflow? This RQ aims at investigating the conceptual and technical concerns in the questions with accepted answers that software developers face in mobile application development.

- **RQ3**: Which are the hot-topics that describe the unanswered questions related to mobile development in Stack Overflow? RQ3 considers unanswered questions, in contrast to RQ2, because we are interested in the most important conceptual/technical concerns that characterize questions without accepted answers.

In our analysis, we distinguished between questions with an accepted answer and questions with non-accepted answers, similarly to the work by Treude et al. [4], which distinguished between successful and unsuccessful questions. The questions posted on SO can have accepted answers (i.e., answers that are verified and accepted by the question owner) or non-accepted answers (i.e., low-voted answers or unrelated ones, which were not validated by the question owner). Our assumption is that questions with only non-accepted answers are of interest to developers, yet they are hard to answer due to several factors, such as emerging technologies, scarce online support, etc., and these questions are good indications for revealing the trend of new technologies and approaches that the community should support.

Consequently, for RQ1 and RQ2, we used the accepted answers as representatives of successful answers in SO, instead of all the answers given to a specific question. For RQ1, we used the answer owners as representatives of the mobile-developers community. Therefore, we analyzed if the contributors in SO provided successful answers to questions labeled in one or more mobile-development-related tag. For RQ2 and RQ3, we used Latent Dirichlet Allocation [3], similarly to the work by Barua et al. [2], to automatically extract the topics that are present in SO questions and answers, and the topic entropy to select the main topics. For each question with accepted answer we built a document with the terms in the question title, question body, and answer body. However, for RQ3 we only used the title and body of questions with non-accepted answers.

A. Data Extraction Process

We used the posts from the latest official SO dump (August 2012) provided as a PostgreSQL dump2 for the MSR 2013 Mining Challenge [5]. Each thread (also called discussion) in SO is composed of a question and a set of answers, and is labeled using predefined or user-defined tags. The contributor who posted the question is called the question owner (QO). One of the answers in the thread that is validated by the QO as the correct answer is called the accepted answer (AA). Thus, there could be questions without an AA, because no answer is selected by the QO as valid. The contributor, who posts the AA, is called the accepted answer owner (AAO).

For our analysis, we considered two types of corpora. The first one, called aa, consists of the question title and body, and the accepted answer body. The second corpus, called naa, contains only the title and body of the questions for which the QO did not mark an accepted answer. Note that we decided to exclude from the naa corpus the answers, because we assumed that the answers not accepted by the QOs do not reflect the same concepts in the corresponding question.

We selected the questions with the following tags T, related to mobile technologies: android, bada, blackberry, iphone and ios, java-me, phonegap, symbian, tizen, webos, and windows-phone. For the ios and java-me tags, we applied regular expressions to exclude the SO tags that contained the tokens ios or java-me but they were unrelated to mobile development, such as iostream, nagios, kiosk-mode, java-melody, java-metro-framework. Using these tags, we built the aa and naa corpora at two granularity levels: (i) a corpus with the union of all the questions labeled with the tags in T, and (ii) a corpus for each tag in T.

All the documents were preprocessed using the following steps: (i) extract text from HTML content using the Java Swing HTML Parser3; (ii) remove all non-white characters except letters and underscore; (iii) split identifiers using the camel case notation; (iv) remove common words4 (i.e., stop-words); and (v) stem words using the Porter algorithm for English. For the splitting step we used the lang package in the Apache Commons project5 version 3.3.1; and for the stop-words removal and stemming we used Apache Lucene Core6 3.6.0. A description of the corpus is provided in our online appendix.

B. Analysis Method for RQ1

To answer RQ1, we computed the number of SO users that posted accepted answers in n mobile-technologies, with n ranging from one to ten. In addition we computed the number of SO users that posted accepted answers in all the possible subsets of size one and two in the power set of the considered tags (e.g., the number of accepted-answer-contributors in Android questions, Android or iPhone questions, etc.).

---

2 http://2013.msrdconf.org/challenge_data/201208_stack_overflow_postgres_dump.tar.bz
3 http://docs.oracle.com/javase/1.4.2/docs/api/javax/swing/text/html/parser/package-summary.html
4 The list of stop-words is included in our online appendix
5 http://commons.apache.org/lang/
6 http://lucene.apache.org/core/
C. Analysis Method for RQ2 and RQ3

We analyzed our corpus using the fast collapsed Gibbs sampling implementation of LDA [6], because it produces equivalent results as standard LDA implementation, yet it is much faster. For more details about LDA we refer the interested reader to [6] [3] [7].

We used the following parameters for LDA: 20 topics (for extracting only the high level topics), 1000 iterations (for convergence), and for the hyper parameters we choose standard values used in the information retrieval community on natural language corpora: \( \alpha = 0.01 \) and \( \beta = 0.01 \). Note that we chose a low \( \alpha \) value because we were interested in high variability among the topic distribution, to easily identify the dominant topics (see below). However, during our evaluation we tried other combinations of LDA parameters and we observed similar results as for the previously enumerated configuration.

For each type of corpora (e.g., aa and nna) consisting of the documents from a specific tag (and all documents), we computed two metrics using the document to topics distribution matrix \( \theta \) generated by LDA. The matrix \( \theta \) has \( K \) number of topics and \( D \) number of documents and each entry \( \theta_{k,d} \) denotes the probability of topic \( k \) pertaining to document \( d \).

The first metric \( TE(k) \), called topic entropy (or topic scattering) [8] [9] of topic \( k \) is generated using the formula:

\[
TE(k) = -\sum_{d=1}^{D} \theta_{k,d} \cdot \log(\theta_{k,d})
\]  

The topic entropy (TE) measures how scattered is a topic across the corpus. For instance, topics with low entropy are concentrated in a small set of documents, whereas topics with high entropy are scattered throughout the corpus. Therefore, we considered topics with high TE values as important (i.e., hot-topics). Note that we did not normalize the TE as in [8] because we did not need to compare the TE across various sized corpora.

The second metric \( NDDT(k) \), called number of documents with dominant topic \( k \) is defined as:

\[
NDDT(k) = \sum_{d=1}^{D} I(d,k), \quad I(d,k) = \begin{cases} 1 \theta_{k,d} = \max_{i=1..K} \{\theta_{i,d}\} \\ 0 \text{i.o.c.} \end{cases}
\]

A topic \( k \) is dominant in document \( d \) if and only if \( \theta_{k,d} \) is equal to the maximum topic probability among all the topics in \( d \). Therefore, \( NDDT(k) \) quantifies the number of documents for which topic \( k \) is the most important.

III. ANALYSIS OF THE RESULTS

In this section we report the results aimed at answering the three research questions formulated previously.

A. Results RQ1

Table 1 lists the number of AAOs, grouped by the number of mobile technologies. Posting answers in only one technology is the predominant choice, with 34,911 contributors that represent the 86% of the sample. Providing answers for two technologies is the choice of 11% of the contributors. Finally, 3% of the contributors provided accepted answers in more than two technologies.

Table 2 lists the number of SO contributors that posted accepted answers in two different technologies. Android, iPhone, and Windows-phone are the top three technologies preferred by the AAOs. According to the developer’s mindshare index reported by VisionMobile [10], 72% of the developers use Android, 56% use iOS, 21% use Windows Phone, 16% use BlackBerry, 7% use Symbian, and 2% use Bada. Our findings show that the order of preferred mobile platforms in SO, is the same than in the developer’s mindshare index [10]: 44.88% of the AAOs prefer Android, 42.07% prefer iOS, 45.00% prefer Windows Phone, 2.55% prefer BlackBerry, 1.89% prefer JME, 0.54% prefer Symbian, and 0.11% prefer Bada. In the case of cross-platform tools, Phonegap is preferred by 3.26% of the AAOs; meanwhile 0.18% of the AAOs provided answers about Webos. Moreover, in the case of multi-platform contributors, the most preferred “duos” are Android-iPhone with 1,599 contributors and Android-BlackBerry with 316 contributors.

Therefore, for RQ1, we conclude that the majority of developers contributing with accepted answers in SO prefer to work in only one mobile platform. However, there are multi-platform contributors that provided accepted answers related to more than one mobile technology.

B. Results RQ2 and RQ3

Columns 2 to 6 of Table 3 address RQ2, and columns 7 to 11 address RQ3. The rows list the TE and NDDT for a subset of topics in both corpora, and five descriptive words (i.e., topic labels) that we selected from the top 15 representative words for the topics. We assigned the topic labels by analyzing the top 15 words related to the topic and by reading the contents of some SO posts with the dominant topic. The complete table is found in our online appendix. We listed the topics ordered descendingly by their TE measure. We can observe that for aa and nna the topics range from a general (i.e., large topic entropy) to a specific scope (i.e., low entropy). For example, topic six, which has the highest entropy (and 29,194 NDDTs) encapsulates the broad need of users to find answers to their
TABLE 3. A SUBSET OF TOPICS AND THEIR REPRESENTATIVE WORDS FOR THE AA AND NAA CORPORA

<table>
<thead>
<tr>
<th>Questions with accepted answers (aa corpus)</th>
<th>Questions without accepted answers (naa corpus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos</td>
<td>k</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
</tr>
</tbody>
</table>

We are currently working on making the study reproducible in TraceLab [11] a framework designed for constructing and sharing experiments in Software Engineering. In addition, we plan to replace the process of choosing default parameters for LDA, by leveraging LDA-GA [12], an approach that automatically finds the best suited LDA parameters that produce a near-optimal LDA model, based on the cohesiveness of the topics. More details about integrating this experiment with LDA-GA and TraceLab are found in our online appendix.

IV. FUTURE WORK

This work is supported in part by the United States NSF CCF-1016868 and NSF CCF-1218129 grants. Any opinions, findings and conclusions expressed herein are the authors’ and do not necessarily reflect those of the sponsors.

REFERENCES