

Fragmentation: A Comparison of Android Vendor's Bugs via Topic Analysis

Dan Han, Chenlei Zhang, Xiaochao Fan, Abram Hindle, Kenny Wong and Eleni Stroulia

Department of Computing Science

University of Alberta

Edmonton, Canada

{dhan3, chenlei1, xf2, hindle1, kenw, stroulia}@cs.ualberta.ca

Abstract—The fragmentation of the Android ecosystem is a topic of substantial debate, and pundits claim (without much actual evidence) that it is worsening. In this study, we search for evidence of fragmentation in the topics of vendor-specific bug reports.

More specifically, we have mined and analyzed the Android bug reports related to two popular vendors, HTC and Motorola. First, we manually annotated the bug reports with labels. Next, we used Labeled-LDA (Latent Dirichlet Allocation) on the labeled data and LDA on the data without manual labels to infer their topics. Finally, by examining the relevance of the top 18 bug topics over time for the bugs reported by each of the vendors, we categorized the topics into three types: *Common Troubled Topics*, *Common Improved Topics* and *Unique Topics*. The *Common Troubled Topics* show that there is no correlation between the troubled features of Android and Android evolution. The *Common Improved Topics* show that some features within the same vendors have portability issues across their multiple devices. The existence of *Unique Topics*, exclusive to each vendor, shows that different vendors have specific bug topics which imply there may be the portability problem on the different vendors. Our findings can be used by Android system community, stakeholders, Android device vendors and developers to make project dashboards, process investigation and feature analysis. In this study we show that there is evidence with Android's bug repository of fragmentation across Android handsets of multiple vendors.

Keywords—Bug reports; Topic mining; LDA; Labeled-LDA; Fragmentation

I. INTRODUCTION

Mobile-device vendors continuously compete against each other for increased market share, and the market landscape is extremely volatile¹. Together, iPhone and Android phones constitute almost 70% of the US mobile-phone market [1]. These two very successful platforms are very different from each other: where Apple tightly controls the software (iOS) and the hardware (iPhone) platform and its evolution, there exist a variety of Android phones produced by different vendors, which often come with custom software, thus giving rise to hardware and software fragmentation in the Android ecosystem [2]. The term *hardware fragmentation* refers to the fact that at any point in time, devices based on the

same Android operating system run on different processors, graphics cards, and screen sizes [2]. The term *software fragmentation* refers to three related problems. First, there exist multiple versions of the Android operating system. Second, vendors offer customized device-specific Android versions. Finally, carriers also offer software customizations.

This fragmentation implies an opportunity for personalization and increased usability, since it enables users to choose the device and software that best meets their needs. On the flip side, it also implies that, due to insufficient cross-platform testing [3], Android applications may not behave consistently across devices and software. Fragmentation tends to delay some users from updating their apps and Android operating system until their specific device is fully supported. These problems may cause users to lose confidence in the Android platform and cost the brand market-share. Given its potential impact, Android fragmentation is the topic of much discussion and controversy. However, there has been little empirical evidence on how fragmentation impacts consumer opinion.

Our objective in this study is to search for evidence of Android fragmentation within the bug reports submitted by users of Android devices from different vendors. To help explore these bug reports, we applied topic analysis on the Android bug reports.

Several topic-analysis methods have been used by researchers in software engineering, including Latent Dirichlet Allocation (LDA) [4], [5], Latent Semantic Indexing (LSI) [6], and Labeled Latent Dirichlet Allocation (Labeled-LDA) [7]. We applied both Labeled-LDA and LDA topic-analysis on the bug reports of different vendors, and identified several recurring topics in these bugs. We also compared the results of Labeled-LDA and LDA to see if they produce similar results even though require very different amounts of manual effort to use. Next, we examined the bug topics to gain some intuition on the causes of Android fragmentation.

Our study focused on the bug reports of HTC and Motorola, two of the most prominent Android phone vendors. The first HTC Android phone was the “HTC Dream”, manufactured in Oct. 2008. HTC has made more than thirty different Android phones since then. Motorola produced their first Android phone in Oct. 2009 and has released more

¹The Global Smartphone Market Landscape: <http://www.asymco.com/2011/11/17/the-global-smartphone-market-landscape> (retrieved March, 2012)

than twenty different and very popular Android phones since then.

This paper makes three important contributions. First, it provides empirical evidence about Android features that contribute to both software and hardware fragmentation. Second, and equally importantly, it lays out a methodology, that can be used to examine fragmentation within systems, such as Android, with numerous hardware platforms. Finally, by applying both Labeled-LDA and LDA we evaluated if the extra effort to label bug reports for Labeled-LDA is worth the effort compared with labeling LDA topics.

The rest of the paper is organized as follows. Section II reviews the background of our work and Section III discusses related work. Section IV introduces our methodology, which is applied to our data set in Section V. Section VI discusses the evidence for fragmentation within Android. Section VII compares and evaluates the topic models generated by LDA and Labeled-LDA. We conclude with threats to validity in Section VIII, and conclusion and future work in Section IX.

II. BACKGROUND

Topic analysis, with respect to *Software Control Systems* (SCS) is useful in a variety of text processing applications [8]. It includes two main steps: topic identification and text segmentation [9]. It can be used in indexing the texts automatically to retrieve information. With it, we can understand what the main topics and sets of associated words with these topics, and where those associated words lie within the text [9]. Recent topic analysis technologies include LDA and Labeled-LDA.

LDA is an unsupervised topic model to credit text documents as mixtures of latent topics, where topics correspond to key word lists presented in the corpus [10]. It has been successfully used in the software engineering area for mining and retrieving informations from large text corpora.

In our research, we apply Labeled-LDA to perform topic analysis. Labeled-LDA is a supervised topic model for credit attribution in multi-labeled corpora [7]. It defines a one-to-one mapping between LDA's latent topics and tags labeled by users. In other words, Labeled-LDA incorporates the multiple tags into the topics learning process and only builds topics around these tags, which is quite different from LDA. LDA, as a totally unsupervised algorithm, automatically learns a set of terms for each topic on a corpus without any constraints. To apply Labeled-LDA, we utilize the Stanford Topic Modeling Toolbox (STMT) [11].

III. RELATED WORK

Topic models have been used to help understand software systems. Marcus et al. [6] used Latent Semantic Indexing (LSI) on both source code and user queries and then identified the most relevant source code documents with similarity measurements. Asuncion et al. [4] applied a coherence measurement on topics learned by LDA to model

the quality of bug reports. Linstead et al. [5] performed LDA to generate traceability links for artifacts in software projects automatically. Topic modeling is also utilized by Thomas et al. [12] to study the evolution of topics in software projects.

Compared with all these approaches, the most important difference is the topic models applied in the studies. They used LDA to extract topics, while we applied Labeled-LDA to obtain the topics. With LDA, researchers need to predefine the number of topics and the topics are often hard to interpret. In our work, we first manually labeled bug reports with multiple labels. Then we employed Labeled-LDA which constrains the topic model to just use those topics that correspond to a document's label set [7]. While during the learning process, LDA cannot incorporate these manual labels. The manual work in our study would overcome the disadvantages of these unsupervised algorithms by pre-defining the number of topics and interpreting the extracted topics.

IV. METHODOLOGY

Our methodology for investigating Android fragmentation starts with extracting bug reports, labeling the bug reports and then applying Labeled-LDA on the labeled bug reports. Then we calculate the average relevance of bug reports to each label over time [13] and compare them between two Android vendors, HTC and Motorola in order to look for fragmentation. We also compare the performance of LDA topics compared with Labeled-LDA topics by comparing the similarity of each pair of topics from LDA and Labeled-LDA.

A. Generating the data

Our first step was to extract the Android bug reports and then find the bug reports relevant to HTC and Motorola. We parse and store the Android bug reports provided by the MSR Mining Challenge [14] as table in a SQL Server database.

Then we selected bug reports that identified themselves as being relevant to HTC or Motorola that mentioned HTC or Motorola in the title text or the description text of the bug report. We then removed all the declined and duplicate bug reports, leaving us with 1503 HTC bug reports and 1058 Motorola bug reports.

B. Creating Labels and Training Annotators

In order to investigate fragmentation from a feature-oriented perspective we needed to label the bug reports by their relevant features. This would allow us to find feature-relevant bug reports for each manufacturer. To ensure our feature-oriented labels would agree with actual Android features we studied various descriptions² of Android's

²Android Operating System summary: http://en.wikipedia.org/wiki/Android_operating_system; Android Market: <https://play.google.com/store/apps>; Android Comparison: http://en.wikipedia.org/wiki/Comparison_of_Android_devices (retrieved March, 2012).

Table I
MANUAL LABELS FROM BUG REPORTS OF HTC AND MOTOROLA.

Vendor	Label
HTC	sms/mms calling email contact video time network android_market display browser bluetooth audio notification image SIM_card settings layout app wifi google_map keyboard calendar alarm language car dialing USB touchscreen CPU gtalk voicemail signal google_voice ringtone google_navigation location font google_earth battery google_translate twitter date VPN picassa video_call rSAP region screen_shot download IPV6 SD_card storage 3G proxy compass calculator synchronize voicemail voice_recognition facebook flash google_latitude GPS camera youtube input search radio system memory upgrade lock
Motorola	calling network settings gtalk calendar signal contact android_market input camera image app wifi keyboard layout sms/mms bluetooth display browser email alarm audio multimedia_dock car SD_card screen voicemail battery upgrade dialing ringtone volume video time swype search exchange headset synchronize facebook google_wave download youtube upload monkey flash VPN touchscreen vibrate CPU system notification text lock GPS calculator USB

operating system, popular apps, and the Android offerings of HTC and Motorola.

Once we became familiar with the Android operating system and Android ecosystem we needed to agree and train ourselves to consistently label Android bug reports.

Following a grounded theory-like coding approach, similar to the approach taken in by Hindle et al. [15], authors Zhang and Fan selected a set of HTC 248 bug reports to label separately.

To label a bug report, the annotator (Zhang or Fan) reads the bug report text, both the title and the description, and then based on their personal interpretation they related that bug report to the relevant features. This means that one bug report can receive multiple labels if it is relevant to multiple identified features. Labels were created as necessary, if a label regarding a feature did not already exist, it was created. These labels consisted of the features and applications on an Android mobile phone, such as SMS/MMS, browser and Wi-Fi or the components of the handsets mentioned in the bug reports, such as GPS, screens and keyboards.

To ensure consistency and agreement in labeling the authors trained themselves in consistent labeling. Each annotator, authors Zhang and Fan, separately labeled each of these 248 bug reports, with labels inspired by the previous research on Android features. Upon completion of labeling, Zhang and Fan compared the labels and discussed label agreement and disagreement in order to train themselves to consistently label bug reports. The topics of the labeled

bug reports were also compared: each annotator’s labeled data was used as input to Labeled-LDA which produced a set of topics. These topics and their relevant bug reports were compared to ensure that annotators had a consistent interpretation of the bug reports and their labels.

C. Labeling the HTC and Motorola Bug Reports

Once the labeling rules were agreed upon each annotator (Zhang and Fan) separately labeled HTC and Motorola bug reports, taking over 60 man hours of manual labeling effort. Using the previously stated labeling methodology, labels were created as necessary. For example, the label “calculator” was created because later in Android’s history there were several bug reports about the correctness of the calculator’s results.

1304 HTC and 985 Motorola bug reports were labeled with multiple labels, leaving 199 and 73 bug reports that cannot be clearly labeled. In total, there are 72 labels for HTC and 57 labels for Motorola. Table I lists all the manual labels from bug reports of HTC and Motorola.

D. Applying Labeled-LDA

Once the bug reports were labeled we wanted to extract the topics associated with the labels. First we had to process the bug reports in order to apply Labeled-LDA to the labeled bug reports. We converted the title and description of each bug report to lowercase, split the text into tokens, and filtered out stop words (words that are less than 3 characters and common English stop words such as “all”, “about”, “the”, “that” and “were”). Then produced word counts/distribution from these sets of bug-report derived words.

Separately, we applied Labeled-LDA to these processed HTC bug reports and Motorola bug reports. Labeled-LDA then outputs the topics, word distributions, associated with our labels, as well as a document-topic matrix which links our labels (topics) to the bug reports from HTC and Motorola.

The topic analysis is based on these results. To visualize the association of a label (an extracted Labeled-LDA topic) to bug reports over time, we grouped all the bug reports by month from 2009 to 2011 based on their open date for each of the two vendors. We then computed the average relevance values of bug reports to this label in each month. The average relevance value of a label l_i in month m_j is the sum of all the relevance values of this label over all bug reports in this month divided by the number of bug reports in this month,

$$A(l_i, m_j) = \frac{\sum_{k=1}^{|m_j|} r(l_i, d_k)}{|m_j|} \quad (1)$$

where $r(l_i, d_k)$ is the relevance value of label l_i to bug report d_k , $|m_j|$ is the number of bug reports in this month. We generated a distribution of average relevance across three years of Android history each label, depicted in Figure 2, Figure 3, Figure 4 and Figure 5.

E. Applying LDA

In order to compare the performance between LDA and Labeled-LDA and to see if Labeled-LDA is worth the manual labeling effort, we applied LDA to the same processed bug reports of HTC and Motorola but without our manual labels.

Applying LDA had one complication, LDA requires an input, n that determines the number of topics that LDA is supposed to extract. If n is too large, the topics tend to repeat themselves and tend to represent similar issues. If n is too small, the topics tend to be cluttered and lack a coherent focus. This can be interpreted manually by reading the topics and evaluating the top 10 or 20 words associated with a topic. To choose the number of topics n , we ran LDA using multiple values of n ranging from 10 to 70, counting by 5, on the bug reports of HTC. Three of the authors (Han, Zhang and Fan) evaluated the word distribution of each topic together in each case of n . We determined if topics were distinct enough based on matching the topics to labels we had created and used for Labeled-LDA. For a given n , if the labels did not repeat too much, and topics did not receive too many labels, then we preferred that n over others without these characteristics. The authors chose $n = 35$, as the topics generated by LDA with $n = 35$ were distinct from each other, had few repetitions and could be interpreted well by the authors based on their own judgment. Other researchers had some similar results [12], [15], [13].

We applied the same process to the bug reports of Motorola and we chose the number of topics to be $n = 30$. As described for the HTC bug reports, we also labeled each topics generated by LDA with our manual labels. Three of the authors annotated the topics together and it took two hours in total to finish all the labeling work. Table II lists a few selected topics from LDA with manual labels.

F. Comparing the Effort to Use LDA and Labeled-LDA

In order to determine if LDA would generate similar results to Labeled-LDA we had to compare the topics of each. Both LDA and Labeled-LDA produce matrices of the relationships between bug reports of two vendors and the labels or topics. Thus we wanted to know if the LDA extracted topics that we manually labeled matched the Labeled-LDA topics that were labeled via our bug report labeling. If the results were similar there would be little point in applying Labeled-LDA in the future.

We determined topic similarity by comparing the sets of documents relevant to a LDA topic and those relevant to a Labeled-LDA topic. Because the LDA topic might be different from the Labeled-LDA topic we did pair-wise similarity comparisons.

We applied the Jaccard similarity coefficient to compute the similarity between each topic in LDA and each label in Labeled-LDA. That is, the Jaccard similarity coefficient between label A in LDA and label B in Labeled-LDA is the

Table II
SELECTED TOPICS FROM LDA WITH MANUAL LABELS. WORD LISTS ARE INFERRED BY LDA.

Vendor	Label	Top 10 terms
HTC	sms/mms	sms, message, text, sent, send, conversation, received, reply, time, number
	email	Email, mail, gmail, app. Inbox, send, emails, message, client, read
	browser	browser, page, web, http, open, website, webview, click, url, load
Motorola	wifi	connect, xoom, hotspot, netbook, wifi, ssid, radio, connection, feature, model
	calendar	calendar, event, sync, appointment, date, google, time, droid, day, change
	contact	contact, google, number, address, list, facebook, droid, account, sync, separate

ratio of the intersection of bug reports related to label A and label B to the union of the bug reports related to label A and label B,

$$sim(A, B) = \frac{\phi(A, d) \cap \phi(B, d)}{\phi(A, d) \cup \phi(B, d)} \quad (2)$$

where the $\phi(A, d)$ is the set of bug reports that has relevance values to label A and d is a set of all the bug reports in each vendor.

The topic-document matrix often contains noise and weak relationships between topics and documents, thus it is necessary to provide a threshold of document relevance to determine if a document is relevant to a topic or not. We used several thresholds (0.01, 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5) on the relevance value of a bug report to a topic in LDA when generating the Jaccard similarity coefficients. We eventually chose 0.2 as the similarities had the biggest mean value. We plotted these pairwise tests (see Figure 6 and Figure 7) in order to explore the match between LDA and Labeled-LDA.

Then we counted the number of bug reports which are related to labels that are both shared by LDA and Labeled-LDA in HTC and Motorola. We applied the Chi-squared test on the two sets of distribution to study if each of the two distributions match.

V. TOPIC MINING AND ANALYSIS

In order to investigate fragmentation within Android, we mined the bug reports of Android and analyzed the results from both quantitative and qualitative aspects.

We started by exploring the distribution of the number of bug reports over time for HTC and Motorola. Then we compared and discussed the distribution of average relevance for each topic over time for both vendors.

A. Overview of bug reports in HTC and Motorola

We grouped the bug reports monthly based on their opened date and counted the total number of bug reports in

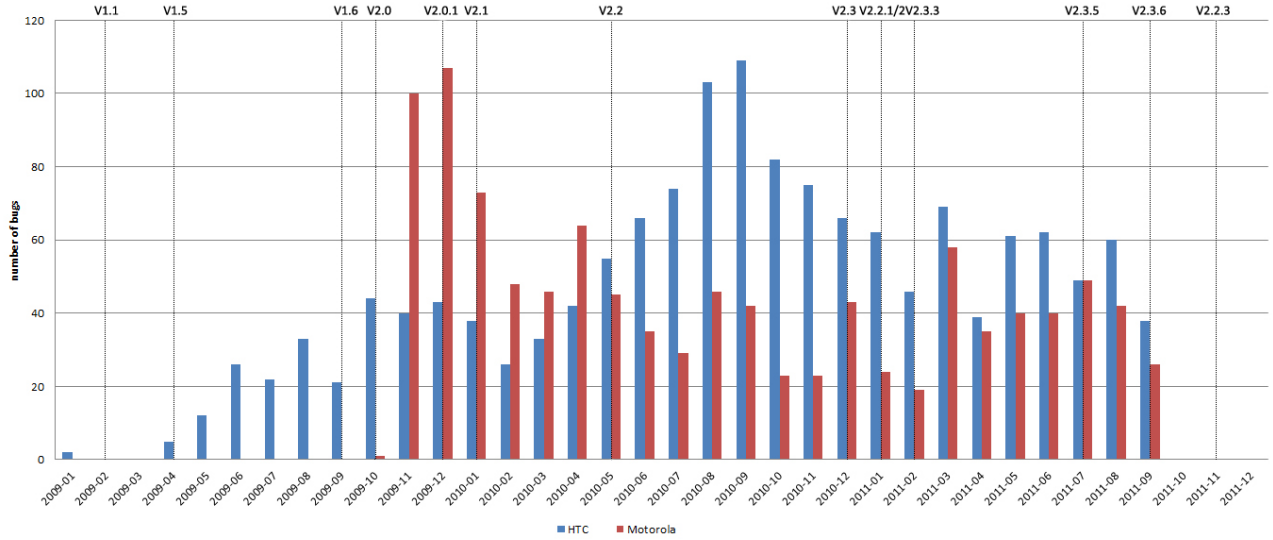


Figure 1. Number of bug reports with the major version of Android for HTC and Motorola

each month for two vendors. Figure 1 depicts a comparison of the number of bug reports for HTC and Motorola.

From Figure 1, we can observe that the first HTC bug report was opened in January, 2009, and the first Motorola bug report was opened in October, 2009. According to the brief history of Android devices survey [16], HTC released the first Android device in October, 2008, while Motorola released its first device in October, 2009. The first bug reports of both vendors are in order of the first device released by them. There is a strong time correlation between the first opened bug report and the first released Android device of both vendors.

In addition, Figure 1 shows the first spike for HTC happened in September, 2010, and for Motorola it happened in December, 2009. By reading the bug reports, we found that the spike of HTC was caused by the fact that many people upgraded Android version from 2.1 to 2.2, and some features did not work well after upgrade. For example, users could not send message after the upgrade. The spike of Motorola mainly contributed to the Android upgrade from 2.0 to 2.0.1. This suggests that the bug reports activities have strong correlation with Android upgrade. It reveals that there might be incompatibility problem among Android versions.

B. Topics Analysis of HTC and Motorola

As shown in Table I we extracted 72 topics for HTC and 57 topics for Motorola with Labeled-LDA.

Based on Equation 1 each topic has a distribution of average relevance over time. According to the comparison of each topic's distribution in both vendors, we categorized the topics into three types, which are *Common Troubled Topics*, *Common Improved Topics*, and *Unique Topics*. The

Common Troubled Topics mean that the distribution of the average relevance of the topics have fluctuations all the time for both HTC and Motorola. The *Common Improved Topics* mean that the distribution of the average relevance of topics turn to be flat over time after several fluctuations for HTC and Motorola. The *Unique Topics* mean that the distribution of average relevance of topics have significant differences between HTC and Motorola.

A representative subset of top 18 topics, which are obtained by sorting the number of related bug reports for HTC and Motorola respectively, is given in Table III. Each topic is associated with top 15 terms generated by Labeled-LDA for both HTC and Motorola. As mentioned before, the label column in Table III represents the feature of Android.

1) *Common Troubled Topic*: Eight *Common Troubled Topics* shared by two vendors are shown in Table III and the distribution of average relevance of each topic is shown in Figure 2. *Common Troubled Topics* shown in Table III for HTC and Motorola share many identical terms. That means they have the same bug reports about sms/mms(text, thread, send), calendar(event, day, google, appointment, time), email(gmail, send, thread), contact(number, google, list), display(screen, button, behavior), bluetooth(headset, connect, calling), synchronize(contact, exchange, google) and settings(turn, network, mode).

We found that multiple topics share some same terms for each vendor. For HTC, five topics including sms/mms, contact, display, bluetooth and settings share the same term "desire". This indicates that these topics happened frequently in HTC Desire device. Calendar and bluetooth share the same term "2.2" and "2.2" means Android version 2.2.

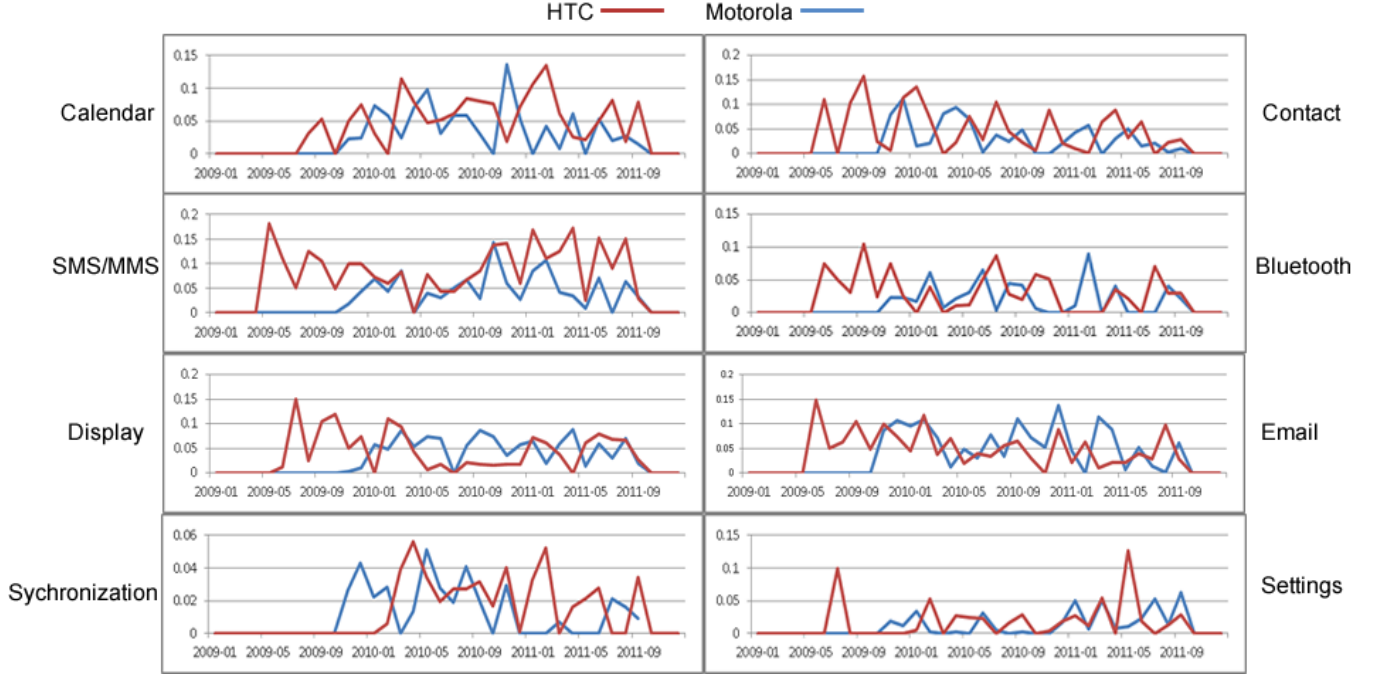


Figure 2. Common Troubled Topics in HTC and Motorola

This indicates that these two topics happened frequently for Android 2.2 in HTC devices. For Motorola, seven topics except *settings* share the same term “droid” and it means Motorola Droid device. In addition, *calendar* and *synchronize* in Motorola share “milestone” which indicates these two topics discussed mostly in Motorola Milestone device. “Xoom” shared by *display* and *settings* indicates that Motorola Xoom has more bug reports related with these two topics. Furthermore, *synchronize* associates with both “Xoom” and “milestone” terms. This indicates bug reports related with *synchronize* happened frequently in both Motorola Xoom and Motorola Milestone.

In Figure 2, HTC and Motorola share the same trend of the distribution of average relevance of topics. Both of them have continuous spikes and drops for each topic over time. There is no obvious decreasing trends of bug reports with Android evolution. In summary, *calendar* in HTC and *display* in Motorola are strongly correlated with different Android versions. *Bluetooth* in both of HTC and Motorola have strong correlation with Android 2.1 and Android 2.2. With Android evolution, these distribution of average relevance of each topic for both vendors do not demonstrate the decreasing trend with Android evolution as we expect. Both of vendors have some topics associated with their typical devices. For HTC, five out of eight topics have correlation with HTC Desire device. For Motorola, seven out of eight topics have correlation with Motorola Droid device. These eight features which correspond to these topics demonstrate the compatibility issues.

2) *Common Improved Topic*: Six *Common Improved Topics* shared by two vendors are shown in Table III and the distribution of the average relevance of each topic is shown in Figure 3.

Common Improved Topics shown in Table III for HTC and Motorola share many identical terms for wifi (*connection,ssid,network*), upgrade (*2.2,2.1,http*), and image(*gallery,picture,photo*). Both vendors have the issues in upgrading from Android 2.1 to Android 2.2. This indicates Android 2.2 might have compatibility issue with the upgrading devices.

Meanwhile, the *Common Improved Topics* also own some special terms. For HTC, bug reports related with *calling* happened frequently in Android 2.1, and bug reports related with *image* and *audio* happened frequently in Android 2.2. For Motorola, bug reports related to *calling* happened frequently in Android 2.2. In addition, four out of six topics have correlation with HTC Desire device. For Motorola, seven out of eight topics have correlation with Motorola Droid device and the other one have correlation with Motorola MileStone device. Therefore, these six topics have strong correlation with the Android hardware devices for each vendor.

In Figure 3, both HTC and Motorola have spikes in the early stage, and then stay in their values. It indicates the corresponding features of Android tend to be more robust over time with Android evolution during the whole observed period.

In summary, *calling* from both vendors has different correlation with Android versions. With the evolution of

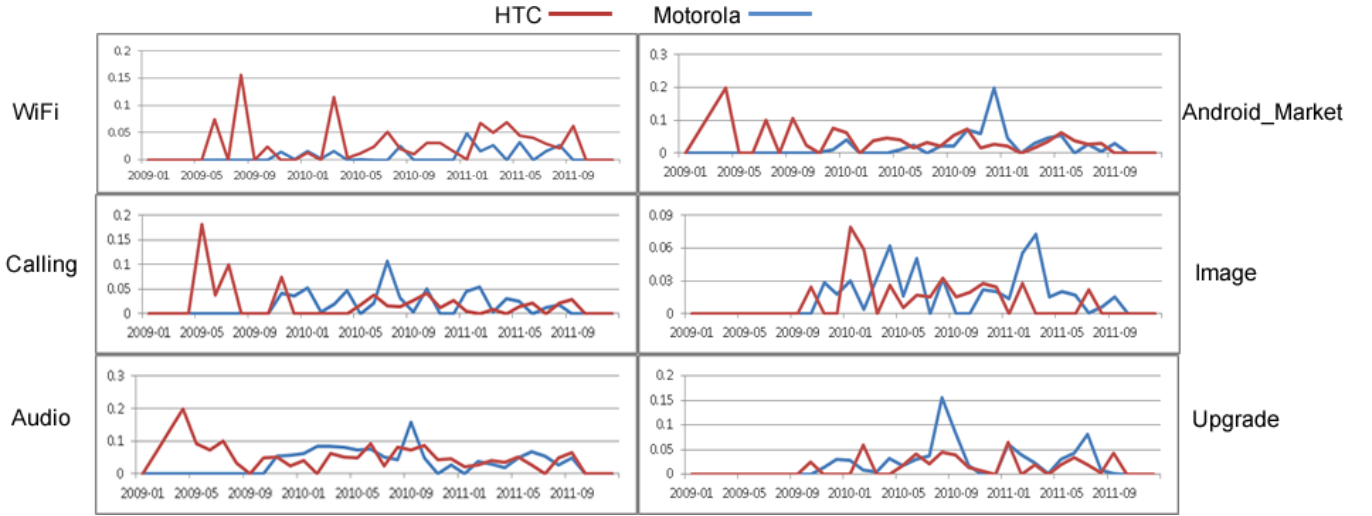


Figure 3. Common Improved Topics in HTC and Motorola

Android, these distributions of average relevance of topics demonstrate the improved trends with Android evolution as expected. However, these topics still have strong correlation with their typical devices for both vendors. This indicates these topics contribute to the hardware fragmentation.

3) *Unique Topics*: There are the two unique topics for HTC shown in Table III. Figure 4 shows the distribution of the average relevance of each topic.

HTC Unique Topics in Table III indicates that HTC has an unique topic which is language (*arabic, desire, language, 2.2, letters, characters, translation, character, read*). The associated terms indicate that bug reports related with language happened frequently in Android 2.2. This stems from the fact that the feature of “keyboard multiple language” is a new feature introduced in Android 2.2. Moreover, most of HTC devices have no physical keyboard, so this new feature has been used frequently by HTC users. In contrast, for Motorola, most of devices have the physical keyboard, so this feature has seldom been used. This fact can also be the reason why HTC has “on-screen” and “virtual” terms for *keyboard*, while Motorola does not have these terms at all.

In Figure 4, HTC *keyboard* turns to stay steady, while Motorola has spikes and drops over time. HTC *language* has the relevance distribution, while there are few bug reports related with *language* to make *language* as a topic in Motorola.

There are two unique topics for Motorola shown in Table III. Figure 5 shows the distribution of the average relevance of each topic.

Motorola Unique Topics in Table III represents HTC and Motorola share the identical terms for GPS (*gps, data, position, location, maps, google, time, lock, wrong, icon, turn, home, latitude*) and browser (*browser, page, text, http, open, server*). Furthermore, both vendors have special terms

for topics separately. For browser, Motorola has *droid, milestone* and *xoom* terms together. This indicates that the *browser* bug reports happened frequently in three Motorola devices and *browser* has portability issue within Motorola Android devices.

In Figure 5, comparing two vendors, we can see the distribution of the average relevance for GPS and browser demonstrate different trends. For HTC, they have strikes and drops in the early stage, and then stay steady. For Motorola, they stay steady and then have strikes and drops afterwards. In summary, for different vendors, they have specific topics which imply there may be portability problems. For a specific vendor, the associated terms of topics implicate that the corresponding features have portability issues across its devices.

VI. DISCUSSION OF FRAGMENTATION

According to the analysis about *Common Troubled Topics*, we can see that there is no strong correlation between the feature evolution and Android evolution. In addition, The topic, upgrade has strong correlation with Android 2.1 and Android 2.2. As there are some features evolution demonstrate stable trends with Android evolution implicated by the *Common Improved Topics*, we can conclude that Android has compatibility issue in some features.

From *Common Improved Topics* and *Unique Topics*, we can see the same topic from different vendors have different correlation, and they have strong correlation with some specific vendors’ devices. These observations reveal that Android has portability issue in some features.

When we refer to Android, we generally mean all Android versions existing in the world which include both Android branches from Android community and that from vendors. In the sense of Android itself, we can see that Android has software fragmentation issue. We also discover that there are

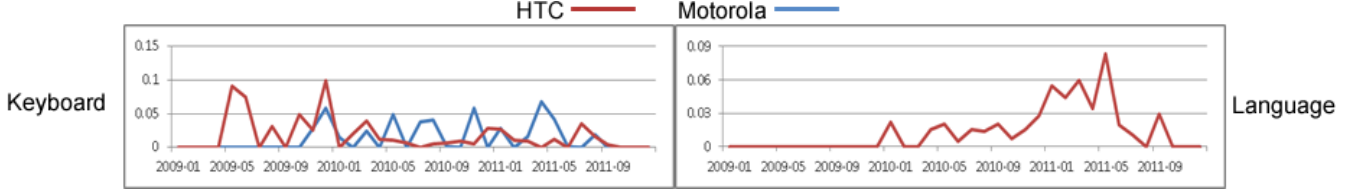


Figure 4. Unique Topics relevance in HTC

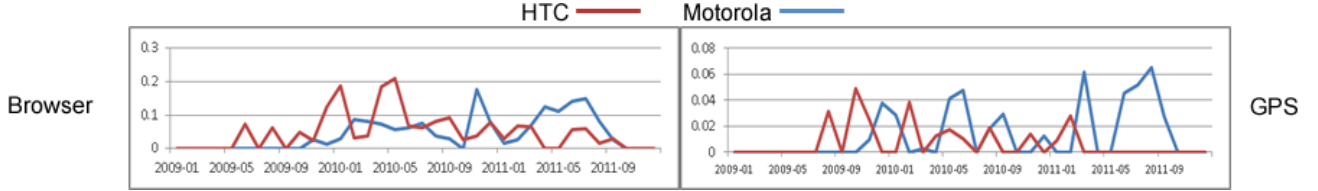


Figure 5. Unique Topics relevance in Motorola

some features has strong correlation with vendors' devices. In the sense of Android devices from different vendor, we can conclude that Android has hardware fragmentation as well.

VII. COMPARING OF LDA AND LABELED-LDA

In this section we investigate if LDA and Labeled-LDA would generate the similar results.

Figure 6 and Figure 7 depict the pairwise Jaccard similarities of labels from LDA and Labeled-LDA. The brighter spots mean the pair of labels have higher Jaccard similarity. These two labels in LDA and Labeled-LDA would be relevant to more similar set of bug reports. The darker spots mean the pair of labels have lower Jaccard similarity and share less bug reports in common.

From these two Jaccard similarity plots (Figure 6 and Figure 7) of labels between LDA and Labeled-LDA, we can observe that most of the Jaccard similarity values are quite small except a few diagonal ones, especially in HTC. This observation is expected since most of the diagonal spots are the Jaccard similarities between the same labels from LDA and Labeled-LDA. However, even the mean similarities of the diagonal spots are just about 0.2 for HTC and 0.08 for Motorola. The similarity plot for Motorola has much more noise than the plot for HTC.

Figure 8 shows the number of bug reports that related to the same labels in the bug reports of HTC and Figure 9 illustrates the number of bug reports that related to the same labels in the bug reports of Motorola. The p values of the Chi-squared test on the two sets of distribution are both close to zero. Hence the number of bug reports related to same labels in LDA and Labeled-LDA are quite different.

We can conclude that only few of the bug reports in HTC and Motorola are predicted by LDA and Labeled-LDA to be related to the same labels. In other words, the relation between topics and each bug report modeled by LDA is

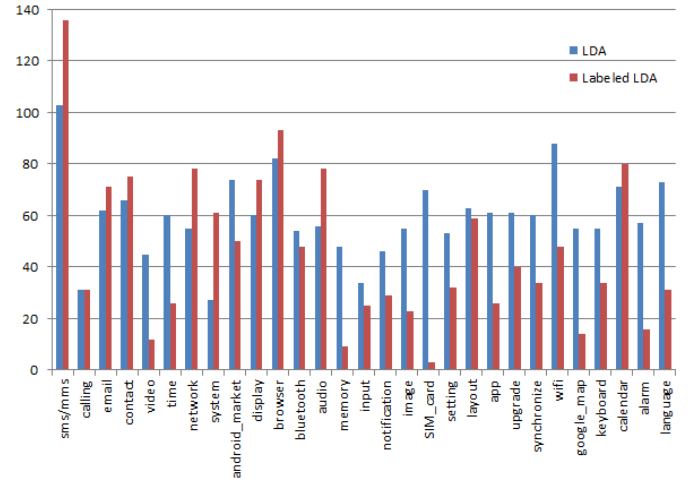


Figure 8. Comparison of number of bug reports related to the same labels from LDA and Labeled-LDA in HTC. The X axis is the same labels from LDA and Labeled-LDA and the Y axis is the number of bug reports.

quite different from the results generated by Labeled-LDA. We think the manual efforts of labeling all the bug reports would help us gain the better topic models generated by Labeled-LDA.

VIII. THREATS TO VALIDITY

Construct validity Our data originated from MSR Mining Challenge [14] and the dataset only ranges from 2009 to 2011. Furthermore we just took all the bug reports related to two vendors in this repository as the dataset to investigate. There may be other bug report repositories can be applied to increase the volume of our dataset.

Internal validity The explanations and theories we built are based on the actual distributions of all the average relevance of labels. The trends in the distributions are just manual observations instead of doing statistical analysis. We

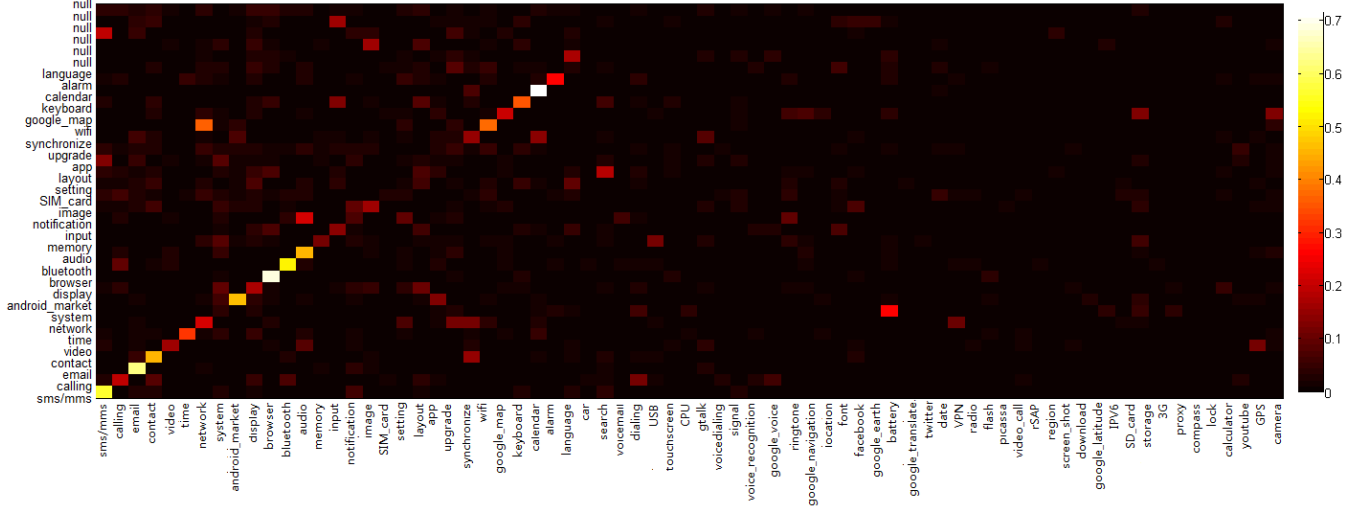


Figure 6. Jaccard similarity of labels between LDA and Labeled-LDA in HTC. X axis is the labels in Labeled-LDA and Y axis is the labels of topics generated by LDA. The label “null” in the Y axis means that topic cannot be labeled. The result is based on the HTC bug reports under the threshold of document relevance of 0.2. Brighter means higher Jaccard similarity.

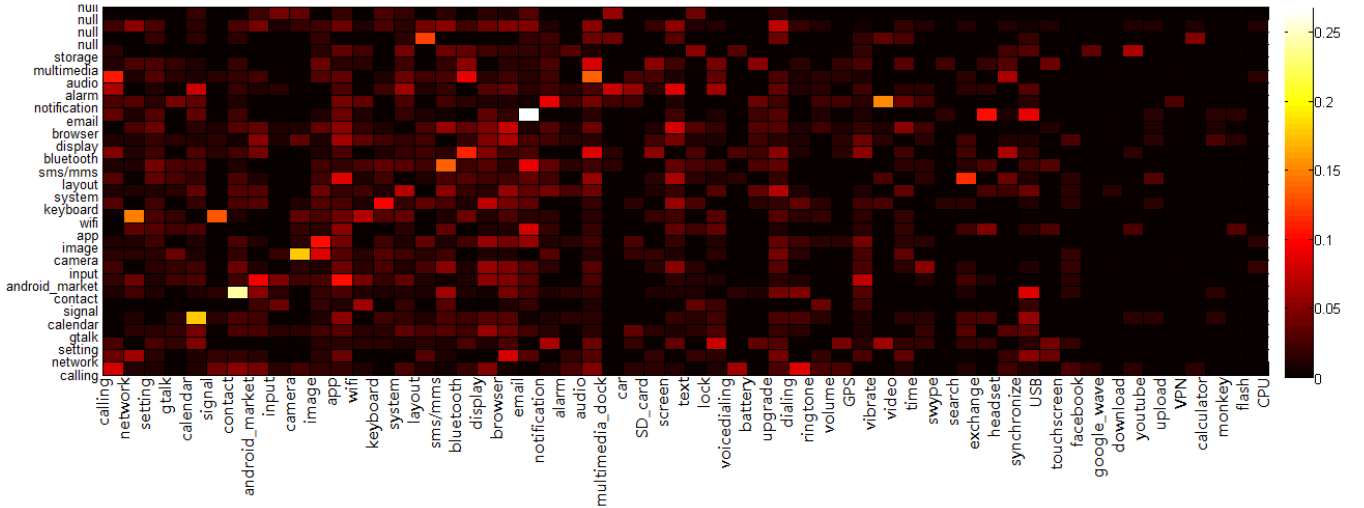


Figure 7. Jaccard similarity of labels between LDA and Labeled-LDA in Motorola. X axis is the labels in Labeled-LDA and Y axis is the labels of topics generated by LDA. The label “null” in the Y axis means that topic cannot be labeled. The result is based on the Motorola bug reports under the threshold of document relevance of 0.2. Brighter means higher Jaccard similarity.

argue that the differences are distinct enough for us to just do observations. Besides, we might suffer from our bias when choosing the terms generated by Labeled-LDA for each label to do analysis.

External validity This study focused on only one project since we cannot find an alternative project that was open source project like Android focusing on mobile platform.

Reliability The labels were from the studying features of Android system by two authors (Zhang and Fan). They cannot hide their previous expertise about Android system and handsets. The labels we come up with might suffer from the biased understanding of the aspects in Android system as

well as mobile devices. Furthermore, when labeling the bug reports, two annotators followed the same protocol and used the same labels. However, they labeled all the bug reports separately. This might affect the labeling consistency in the dataset.

IX. CONCLUSION AND FUTURE WORK

In this paper we studied Android bug reports for two vendors, HTC and Motorola. Based on topic analysis using Labeled-LDA on a corpus of manually tagged bug reports with multiple labels, we extracted the top 18 topics and categorized them into *Common Troubled Topics*, *Common*

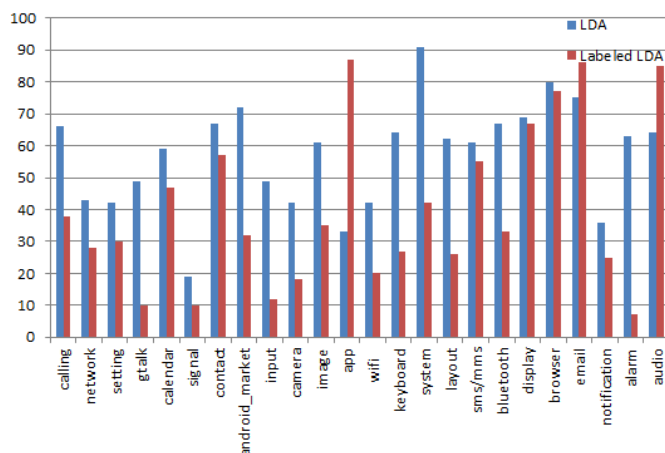


Figure 9. Comparison of number of bug reports related to the same labels from LDA and Labeled-LDA in Motorola. The X axis is the same labels from LDA and Labeled-LDA and the Y axis is the number of bug reports.

Improved Topics and *Unique Topics* for both vendors. The *Common Troubled Topics* show that there is no correlation between the troubled features of Android and Android evolution. In other words, there may be the incompatibility problem existing to the specific features of Android. The *Common Improved Topics* show that some features within the same vendors have portability issues across their multiple devices. The *Unique Topics* show that different vendor has specific bug topics which imply there may be the portability problem on the different vendors. Furthermore, we found that the manual efforts of labeling all the bug reports would help us gain the better topic models generated by Labeled-LDA after comparing LDA and Label-LDA.

For our future work, we will use the name of each hardware model as a label to do topic analysis while applying our methodology in order to discover the effects of different Android versions with respect to compatibility and stability. We will plan to investigate more vendors in order to reveal vendor specific bug topics.

REFERENCES

- [1] "Android And Apple Now Command Nearly 70 Percent Of U.S. Smartphone Market Share," <http://techcrunch.com/2011/08/30/android-apple-70-percent-smartphone>, retrieved March, 2012.
- [2] "An Analysis of Android Fragmentation," <http://www.tech-thoughts.net/2012/03/analysis-of-android-fragmentation.html>, retrieved March, 2012.
- [3] "Zipline CEO: Stop whining about Android fragmentation and do some damn QA!" <http://thenextweb.com/google/2012/04/02/zipline-ceo-stop-whining-about-android-fragmentation-and-do-some-damn-qa/>, retrieved March, 2012.
- [4] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, "Software traceability with topic modeling," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineer-*
- ing - Volume 1, ser. ICSE '10. New York, NY, USA: ACM, 2010, pp. 95–104.
- [5] E. Linstead and P. Baldi, "Mining the coherence of gnome bug reports with statistical topic models," in *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories*, ser. MSR '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 99–102.
- [6] A. Marcus, A. Sergeyev, V. Rajlich, and J. I. Maletic, "An information retrieval approach to concept location in source code," in *In Proceedings of the 11th Working Conference on Reverse Engineering (WCRE 2004)*. IEEE Computer Society, 2004, pp. 214–223.
- [7] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, August 2009, pp. 248–256.
- [8] A. Hindle, M. Godfrey, and R. Holt, "Whats hot and whats not: Windowing developer topic analysis." ICSM.
- [9] H. Li and K. Yamanishi, "Topic analysis using a finite mixture model," *Information processing & management*, vol. 39, no. 4, pp. 521–541, 2003.
- [10] "Latent Dirichlet allocation," http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation, retrieved March, 2012.
- [11] "Stanford Topic Modeling Toolbox:," Website, <http://nlp.stanford.edu/software/tmt/tmt-0.4>, retrieved March, 2012.
- [12] S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein, "Modeling the evolution of topics in source code histories," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, ser. MSR '11. New York, NY, USA: ACM, 2011, pp. 173–182.
- [13] A. Hindle, C. Bird, T. Zimmermann, and N. Nagappan, "Relating requirements to implementation via topic analysis: do topics extracted from requirements make sense to managers and developers," unpublished.
- [14] E. Shihab, Y. Kamei, and P. Bhattacharya, "Mining challenge 2012: The android platform," in *The 9th Working Conference on Mining Software Repositories*, 2012, p. to appear.
- [15] A. Hindle, N. A. Ernst, M. W. Godfrey, and J. Mylopoulos, "Automated topic naming to support cross-project analysis of software maintenance activities," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, ser. MSR '11. New York, NY, USA: ACM, 2011, pp. 163–172.
- [16] "A brief history of Android," http://reviews.cnet.com/8301-19736_7-20016542-251/a-brief-history-of-android-phones, retrieved March, 2012.

Table III
TOPICS AND ASSOCIATED WORD LIST WITH RELATED TOP 15 TERMS

Topic Type	Label	HTC	Motorola
Common Troubled Topics	sms/mms	message, sms, text, thread, time, sent, desire, contact, new, number, conversation, send, version, app, screen	message, text, sms, droid, send, thread, messaging, sent,user, version, version, person, threads, number, http
	email	email, mail, gmail, app, message, inbox, messages,client,emails, account, send, interface, thread, time, new	email, droid, account, gmail, mail, server, message,user,emails, exchange, file, version, open, device, app
	calendar	calendar, event, day, events, google, view, 2.2,time,month, date, version, reminder, appointment, edit, running	calendar, event, droid, google, appointment, events, day, field, date, appointments, outlook, milestone, data, app, version
	contact	contact, contacts, number, freed, activity, displayed, list, group, google, numbers, starting,desire, user, version, field	contact, contacts, droid, number, numbers, address, version, google, menu, correct, behavior, different,list, option, gmail
	display	screen, version, desire,behavior, app, home, number,code, final, press, sure, user, black, new, power	droid, screen, button, correct, home, display, behavior, landscape, 2.1, menu, bar, xoom,device, user, status
	bluetooth	bluetooth,headset, car, connect, device, connection, version, data, app, desire, desire, 2.2, work, connects, behavior,2.1	bluetooth, headset, droid, device,connected, connect, devices, calls,car,issue, connection, 2.2, car,pair, time
	synchronize	contacts, account, sync, exchange, contact, google, ears, device, group, server, Gmail, policy, new, list, display	sync, google, account, contacts, device, contact, group, time, exchange, contacts, display, groups, list, droid, milestone
	settings	volume, sound, set, pattern, default, turn, desire, static, control, apps, change, settings, media, dns, screen	settings, device, menu, turn, network, vpn, honeycomb, button, xoom, settings, behavior, right, wireless, headset, mode
Common Improved Topics	wifi	wifi, access, network, connection, connect, router, ssid, desire, http, wi-fi, device, connected, scan, point, app	wifi, xoom, connect, hotspot, turn, connection, ssid, radio, error,signal, state, user, time, feature,hotspots
	upgrade	update, 2.2, file, 2.1, google version, error, upgrade, froyo, install, work, desire, ota, card, ssl	update, droid, 2.1,2.2, home, http, version, user, issue, device, longer, settings, performance, issues, updated
	audio	music, audio, player, file, play, 2.2,sound, version, time, playing, playback, app, start, reproduce, mp3	music, droid, player, media, audio, files, volume, play, playing, version, app, issue, mode, running, genre, sound, user
	calling	number, calls, calling, 2.1, receive, called, button, answer, bluetooth, desire, screen, incoming, works, time, magic	droid, calls, number, end, button, answer, incoming, screen, voice, speaker, speaker, 2.2, device, place, headphones
	android market	market, app, google, account, download, update, application, user, device, version, apps, paid, desire, installed, application	market, apps, app, device, application, update, open, user, version, time, reproduce, download, purchase, google, milestone
	image	image, gallery, picture, matrix, photo, null, camera, pictures, version, steps, 2.2, photos, code, display, view	image, droid, wallpaper, gallery, photo, picture, device, file, select,video, folder, load, live, stock, size, screen
HTC Unique Topics	language	arabic, desire, language, 2.2, letters, character, translation, character, read, support, sms, write, hebrew, devices,2.3	NONE
	keyboard	keyboard, input,text, key, version, number, typing, on-screen, mode, field, landscape, virtual, keys, type, message	keyboard, droid, keys,text, press, space, box, open, device, key, app, software, 2.0.1, landscape
Motorola Unique Topics	GPS	gps, data, position, location, maps, google, time,lock, wrong, icon, turn, home, latitude, unit, tag, available	maps, gps, google, app, droid, location, application, navigation, map,device, traffic, time, upgrade, turn, route
	browser	browser, page, text, http, open, server,verion, desire, client, web, application,2.1, device, button, user	browser, droid, page, web, http, open, xoom, html, behavior, running, links, issue, milestone, 3.1,text