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

Dan Han, Chenlei Zhang, Xiaochao Fan, Abram Hindle, Kenny Wong, Eleni Stroulia
Department of Computing Science
University of Alberta
Edmonton, Canada
{*dhan4, chenlei1, xf2, hindle1, kenw, stroulia*}*@cs.ualberta.ca*

***Abstract*—Android fragmentation has been a controversial topic, but both proponents and opponents cannot provide strong evidences to support their statements. In order to make the debate more clear, we mined and analyzed the Android bug reports related to two popular Android vendors, HTC and Motorola. We manually annotated bug reports with labels and applied Labeled Latent Dirichlet Allocation (LDA) to the datasets to produce bug topics. By comparing the average relevance of top 20 bug topics over time for both vendors, we categorized the topics into two types which are common topics and unique topics. We investigated and discussed these two types of bug topics relevance tendency over time. Our analysis results lead to the conclusion that Android fragments into multiple incompatible and brand-specific versions. Our findings can be used by Android system community, stakeholders, Android device vendors and developers to make project dashboards, process investigation and feature analysis.**

***Keywords*-Bug reports; Topic mining; Labeled LDA**

I. Introduction

The market share of mobile phones is always increasing and getting more and more competitive among various mobile device vendors[1]. iPhone and Android phone share the majority of mobile device market. Compared to Apples closed ecosystem for iOS, Android system has many fragmentation. These fragmentation highlights include Android five major versions, diverse customized Android versions and various Android mobile devices provided by different device vendors [1].

Android fragmentation has been a controversial topic which swells up now and again. Some people argue that there should be many issues about the fragmentation on Android platform. They hold the point that there are too many existing Android versions, various Android mobile devices running these versions and too many branches from multiple vendors [1][2]. On the other hand, there is an opposite viewpoint that there is no issues about fragmentation in Android community, only differentiation [3]. However, both proponents and opponents cannot provide strong evidences to support their statements.

In order to make this debate more clear, we will explore the bug topics for different vendors from Android bug reports using Labeled LDA [1]. We choose the bug reports of HTC and Motorola in this study. HTCs first Android phone was the HTC Dream manufactured in Oct. 2008. HTC has made more than thirty different Android phones since then. Motorola made their first Android phone in Oct. 2009 and has released more than twenty different Android phones since then . Their Android products have gained widespread popularity.

In this study, our goal is to investigate the topics in bug reports related to the two Android vendors, HTC and Motorola, with the purpose of understanding how their bug topics evolve over time. We utilize the Labeled LDA, a novel technique in software engineering, to build the topic models and analyze the topic evolution models[4] of bug reports in HTC and Motorola. Researchers have applied LDA and other topic models to a lot of software projects[5][6][7][8]. We will also apply LDA on our data and try to evaluate the performance between LDA and Labeled LDA.

This paper makes the following contributions:

- We introduce the Labeled LDA to build topic models.
- We empirically compare the performance of LDA and Labeled LDA on the bug reports of HTC and Motorola.
- We analyze the topic evolution models in Android platform for HTC and Motorola by mining their bug reports.
- Our findings support that Android community does have fragmentation issues.

The paper is organized as follows: Section 2 describes the background; we discuss the related work in section 3; in section 4, we explicate our methodology about the mining approaches applied on this study; section 5 is to compare and evaluate the topic models generated by LDA and Labeled LDA; we introduce the analysis of topic evolution models in section 6; the paper concludes with a discussion of two research questions, threats to validity, conclusion and future work in section 7, 8 and 9 correspondingly.

II. Background

In our research, we apply Labeled LDA to perform topic analysis. Labeled LDA is a supervised topic model for credit

[1]http://asymco.com/2011/11/17/the-global-smartphone-market-landscape

attribution in multi-labeled corpora[9]. It defines a one-to-one mapping between LDAs 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)[10]. Specifically, this tool outputs a set of topics, each one consisting of a list of terms, and the relevance distribution between each bug report and all the topics.

III. Related Work

Topic models have been used to help understand software systems. Marcus et al.[5] 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.[6] applied a coherence measurement on topics learned by LDA to model the quality of bug reports. Linstead et al.[7] performed LDA to generate traceability links for artifacts in software projects automatically. Topic modeling is also utilized by Thomas et al.[8] to study the evolution of topics in software projects.

Compared with all these approaches, our work differs from them in two aspects. We manually labeled bug reports with multiple labels. And we applied labeled LDA in our work to overcome the disadvantages of these unsupervised algorithms by pre-defining the number of topics and interpreting the extracted topics [6].

IV. Methodology

Our methodology is to extract bug reports, assign multiple labels to each of them and then perform Labeled LDA on the labeled data. In order to compare the performance between LDA and Labeled LDA, we also apply LDA on the extracted bug reports of HTC and Motorola without our manual labels. We label all the topics generated by LDA. For each vendor, we calculate the similarity of each pair of labels from LDA and Labeled LDA to evaluate their performance. After that we calculate the average relevance of bug reports to each label over time[11] and compare them between two Android vendors, HTC and Motorola.

A. Generating the data

We use the Android bug reports provided by the MSR Mining Challenge [12]and store all the XML data into a SQL Server. Using regexes, we extract bug reports highly related to Android phones of HTC and Motorola respectively. We use the title, opened date and description in each bug report. After removing all the declined and duplicate bug reports, we have 1503 HTC bug reports and 1058 Motorola bug reports.

Table I

Selected Manual labels from bug reports of HTC and Motorola. There are at least 20 bug reports are assigned to these labels.

Vendor	Label
HTC	sms/mms browser calendar network audio contact display email system Layout android_market bluetooth wifi keyboard synchronize setting calling language time app input image dialing search
Motorola	app email audio browser display contact upgrade screen sms/mms calendar system calling synchronize image GPS android_market setting network keyboard lock Layout notification dialing wifi bluetooth

B. Multi-labeling

We manually tag bugs with multiple labels. The labels are defined as the functionalities 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, screen and keyboard. Most of the labels about functionalities and applications are based on the detailed introduction to the features, specifications and applications of Android operating system[2]. The labels about components of the handsets are mostly extracted from the description of comparison of Android devices[3].

First, two of us (Zhang and Fan) separately tag the HTC 248 bug reports in 2009 with the multiple labels. Then we train each authors labeled data with STMT. By comparing each set of trained topics in the results and returning to the bug reports, we check to ensure the same interpretation of each bug when labeling. Then we come up with the same labeling rules to process the data. After this, we tag the rest of bug reports separately for both HTC and Motorola. When there is a bug report that cannot be tagged by the labels we already have, we would create a new label together based on our definition of labels, i.e. the functionalities and applications on an Android mobile phone or the components of the handsets. For example, the label “calculator” is created by this way. At first we could not expect this basic application on mobile phones would have bugs. However, there are several bugs that complained the correctness of the calculator’s results. So we added it to our labels. The manual labeling took approximately 30 hours per person. In the end, for HTC and Motorola there are 199 and 73 bug reports that cannot be labeled respectively. We have tagged 1304 HTC and 985 Motorola bug reports, each with multiple labels. In total, there are 72 labels for HTC and 57 labels for Motorola. Table I selects Manual labels from bug reports of HTC and Motorola. There are at least 20 bug reports are assigned to these labels.

[2]http://en.wikipedia.org/wiki/Android_operating_system
[3]http://en.wikipedia.org/wiki/Comparison_of_Android_devices

C. Applying Labeled LDA

To apply Labeled LDA we preprocessed the labeled bug reports. We convert the title and description of each bug report to lowercase and remove stop words (words that are less than 3 characters and common English stop words such as "all", "about", "the", "that" and "were"). By applying Labeled LDA to the bug reports of HTC and Motorola separately, we have the word distribution of each label and a matrix that provides the relationship between bug reports and the labels. The topic analysis is based on these results. In order to investigate the trend of each label by time, we grouped all the bug reports by month from 2009 to 2011 based on their open date for each of the two vendors. For each label, we 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. This metric is to sum all the relevance values. At last we generated a distribution of average relevance among 36 months for each label.

D. Applying LDA

In order to compare the performance between LDA and Labeled LDA, we also apply LDA on the extracted bug reports of HTC and Motorola without our manual labels. We used the same preprocessing method on the HTC bug reports when applying LDA.

To choose the number of topics n, we ran LDA using multiple values of n that included: 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65 and 70 on the bug reports of HTC. Three of the authors evaluated the word distribution of each topic together in each case. We then chose the n is equal to 35 where the topics generated by LDA were distinct enough from each other, had few repetitions and could be interpreted well by the authors. Other researchers had some similar results [8] [11]. We did the same process on the bug reports of Motorola and we chose the number of topics is equal to 30. We also labeled each topics generated by LDA for both vendors and the labels are based on 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.

E. LDA vs. Labeled LDA

Both of the results from LDA and Labeled LDA are the matrices that provide the relationship between bug reports of two vendors and the labels. We want to explore whether LDA would generate the similar results to Labeled LDA.

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

That is if the topics generated by LDA that were labeled as the same ones in Labeled LDA would be related to similar bug reports. The Jaccard similarity coefficient is applied 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 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,

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

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.

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 label in LDA when generating the Jaccard similarity coefficient and found that under the threshold is 0.2 the result is best.

V. PERFORMANCE OF LDA AND LABELED LDA

We investigate the performance between LDA and Labeled LDA in this section.

From these two similarity plots(Figure 1 and Figure 2) of labels between LDA and Labeled LDA, we can observe that most of the topic-label similarity values are quite small except the 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. Then we counted the number of bugs that are related to labels which are both shared by LDA and Labeled LDA. That is showed by Figure 3 and Figure 4.

Most of the same labels from LDA and Labeled LDA have the comparable amount of bug reports. For example, the label "calling" from the HTC bug reports has exactly the same number of bugs related to for both results of LDA and Labeled LDA. However, the similarity of these related bug

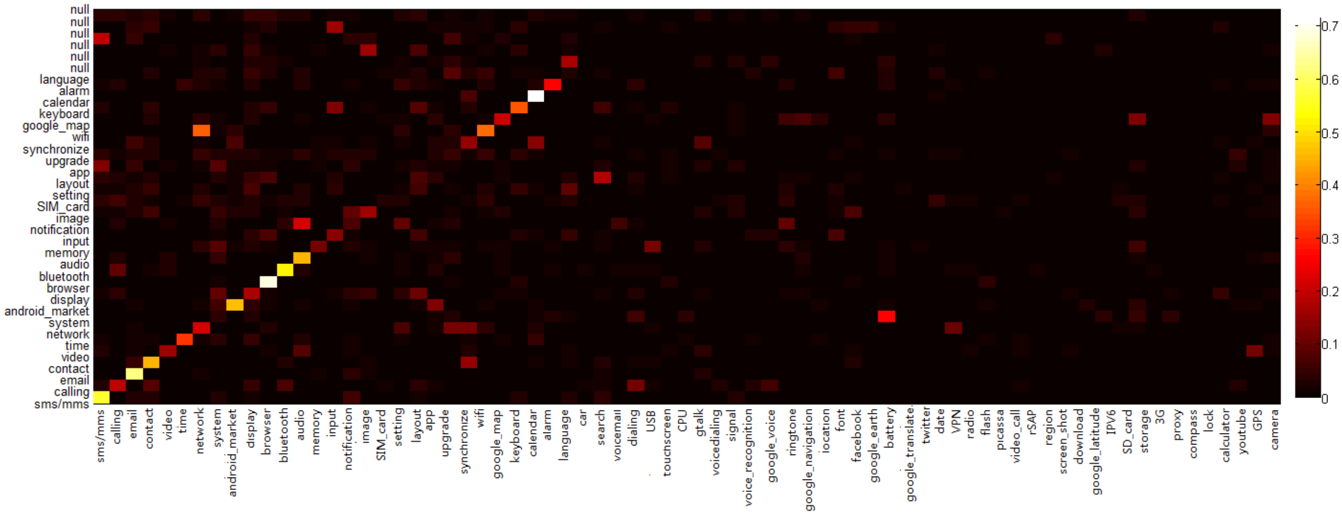

Figure 1. Similarity of labels between LDA and Labeled LDA in HTC. X axis is the labels in Labeled LDA and Y axis is the labels of topics generated by LDA. The result is based on the HTC bug reports. Brighter means higher similarity.

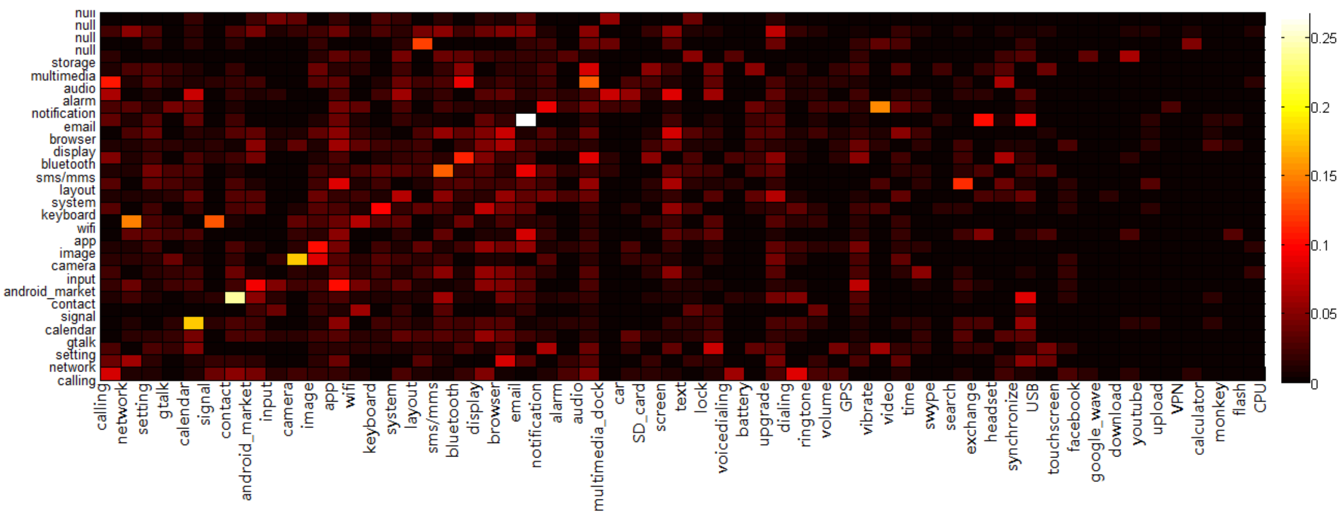

Figure 2. Similarity of labels between LDA and Labeled LDA in Motorola. X axis is the labels in Labeled LDA and Y axis is the labels of topics generated in LDA. The result is based on the Motorola bug reports. Brighter means higher similarity.

reports in terms of this label "calling" is very low which means LDA and Labeled LDA related quite different bug reports to this label. When doing this comparison, we cannot ignore the number of bugs that related to each label from both two techniques. That is, for one label, the ratio (the smaller number is divided by the bigger number so the ratio is always less or equal to one) of the number of bug reports related to this label predicted by LDA to that of Labeled LDA would be the upper bound of the similarity value. From Figure 3 and Figure 4 that the relation between topics and each bug report modeled by LDA is quite different from the results generated by Labeled LDA.

The similarity values for these labels in Figure 2 are quite low compared with the ratio. Only about ten labels in HTC have similarity values that are larger than half of the ratio. For Motorola, the similarity values are all very low compared with the upper bound of the similarity values.

From all above, we can conclude that only a 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 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. Hence we chose to use the Labeled LDA to do topic analysis in the following sections.

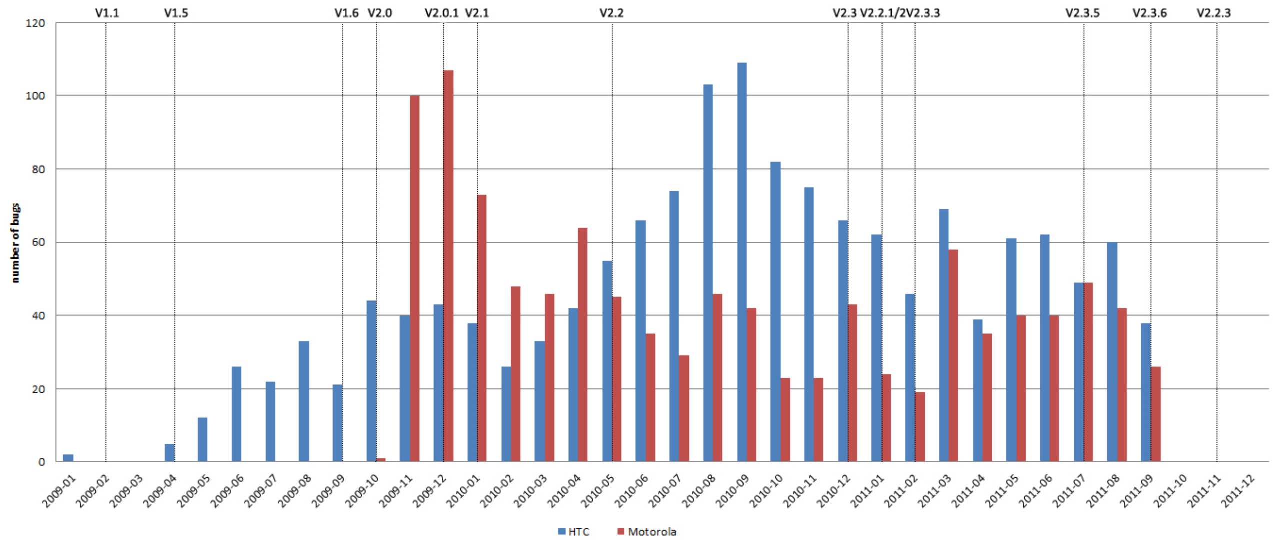

Figure 7. Number of bugs with the major version of Android for HTC and Motorola

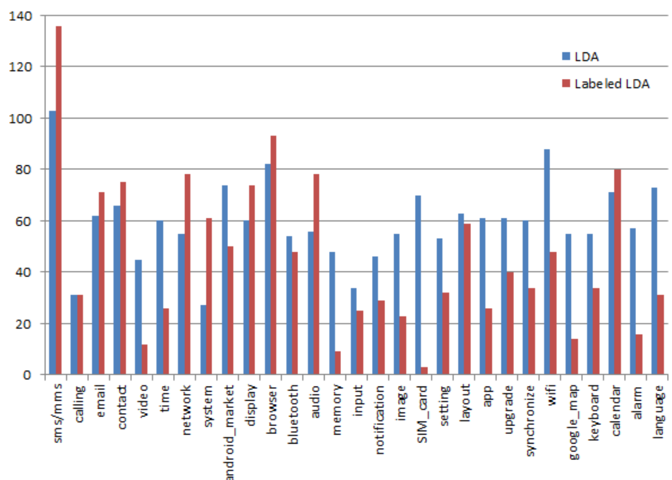

Figure 3. Comparison of number of bug reports related to the same labels from LDA and Labeled LDA in HTC. The X axis is the same labels from LDA and Labeled LDA and the Y axis is the number of bug reports.

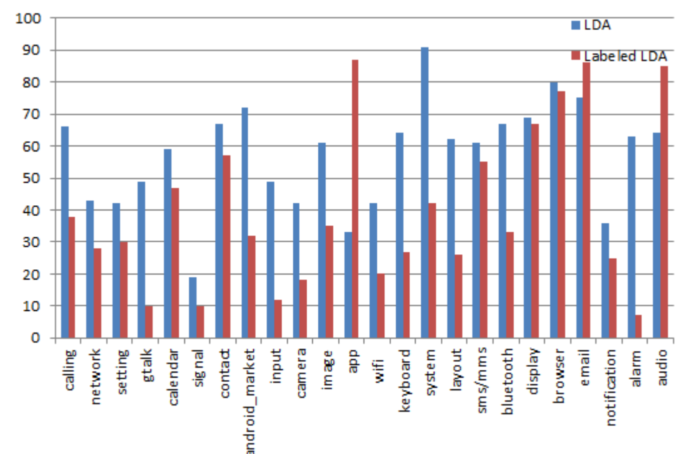

Figure 4. Comparison of number of bug reports related to the same labels from LDA and Labeled LDA in Motorola. The X axis is the same labels from LDA and Labeled LDA and the Y axis is the number of bug reports.

VI. Topic Mining and Analysis

In order to investigate the compatibility and stability of fragmentation in Android system, we mined the bug reports of HTC and Motorola and analyzed the results from both quantitative and qualitative views.

This section describes the results of topic analysis in HTC and Motorola.

A. Overview of bug reports in HTC and Motorola

We start by exploring the quantity of bug reports for HTC and Motorola. We group the bugs monthly based on their opened date and count the total number of bugs in each month for two vendors. Figure 7 depicts a comparison of the quantity of bugs for HTC and Motorola.

Observing Figure 7, we can proceed to answer the following questions:

- Is there any time correlation between the first opened bug reports and the first released Android phone of both vendors?
  Figure 7 shows the first HTC bug report was opened in January, 2009, and the first Motorola bug report was opened in November, 2009. According to the brief history of Android phone survey [13], HTC released the

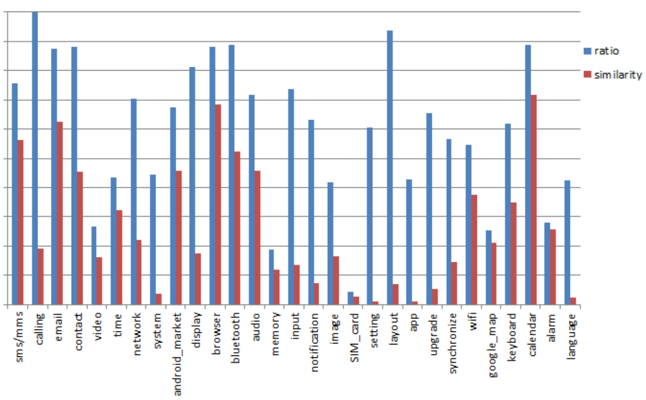

Figure 5. The comparison of ratio and similarity in HTC. The result of the smaller number of bug reports related to this label in LDA or Labeled LDA divided by the larger one is the ratio of this label. The X axis is the same labels from LDA and Labeled LDA.

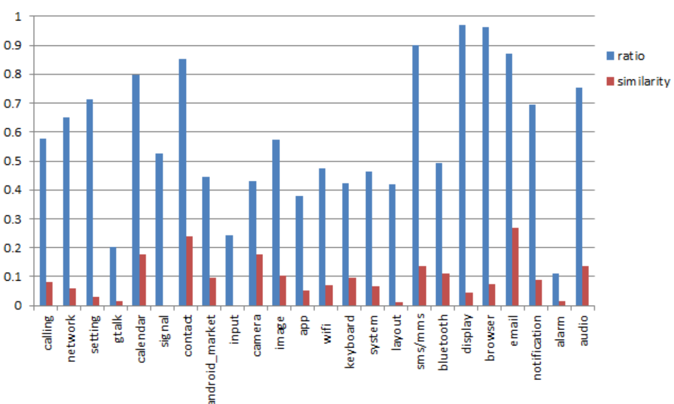

Figure 6. The comparison of ratio and similarity in Motorola. The result of the smaller number of bug reports related to this label in LDA or Labeled LDA divided by the larger one is the ratio of this label. The X axis is the same labels from LDA and Labeled LDA.

first Android phone in October, 2008, while Motorola releases the first phone in October, 2009. The first bug report of both vendors is in order of the first phone released by them.

- Do both vendors have the same market reaction?
Comparing the bug reports distribution and Android phone release history for HTC, there is a one-year time span between the first Android phone released in October, 2008 and the peak number of bug report in October, 2009. For Motorola, there is only a one-month time span between the first Android phone release in November, 2009 and the peak number of bug reports in November, 2009. The longer market reaction is due to the debute of each new technology should undergo the course of recognition. The observation that Motorola reached its peak value of bug reports in such a short time springs from Android system has already been recognized by smart phone market. So it easily brought about the boom period with the fruit from HTC.
- Do both vendors have the same bug reports distribution over time?
Examining the bug reports distribution of HTC over time, we can see there are two notable ups and downs. The watershed of these two waves was February, 2010. The date was close to the release time of HTC Nutex One, which was the first HTC phone based on Android 2.1 announced in January, 2010[13]. All phones released before this date was based on Android 1.x. So each wave indicates the bug reports distribution for one Android major version. The first wave was mainly for bug reports in Android 1.5, and the second one was for Android 2.1. These two waves with similar tendencies indicated that the bug reports distribution of each Android version experienced the same life cycle in HTC Android phones. For Motorola, there was no regular spikes and drops like HTC. The different bug reports distribution would be caused by the different Android phone support. Comparing with the shed of Android 1.x and 2.x supported by HTC, Motorola announced its Android phones with Android 1.5 and 2.x at the same time. With two major versions supported in the same period, the bug reports distribution seemed to be smooth over time.

B. Topics Analysis of HTC and Motorola

First, it should be mentioned that we got 72 topics for HTC and 58 topics for Motorola with Labeled LDA. A representative subset of top 20 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 described by top 10 terms associated with the topic concept for both HTC and Motorola. These topics cover 85% bugs of HTC, and 83% bugs of Motorola.

From Table III, we can see that HTC and Motorola share many identical terms for each label. For instance, we found both vendors have the same terms for Bluetooth such as connection, headset and calling issue. For HTC, this topic happened frequently for the Desire model on both Android 2.1 and 2.2 versions. For Motorola, Bluetooth bugs happened frequently for the Droid with Android 2.2.

Examining the label column of Table III, it is clear that various functions of the Android phone are represented. With the assistance of these associated terms for each topic, we come up with a concrete analysis for each topic and investigation of the relations among the topic relevance, the Android phone released of each vendor and the Android system evolution.

By analyzing the topic distribution among bug reports, we got the average relevance for each topic by month. Examining the average relevance trends for each topic and comparing them for both vendors, we categorize the topics into two types. They are common topic and unique topic.

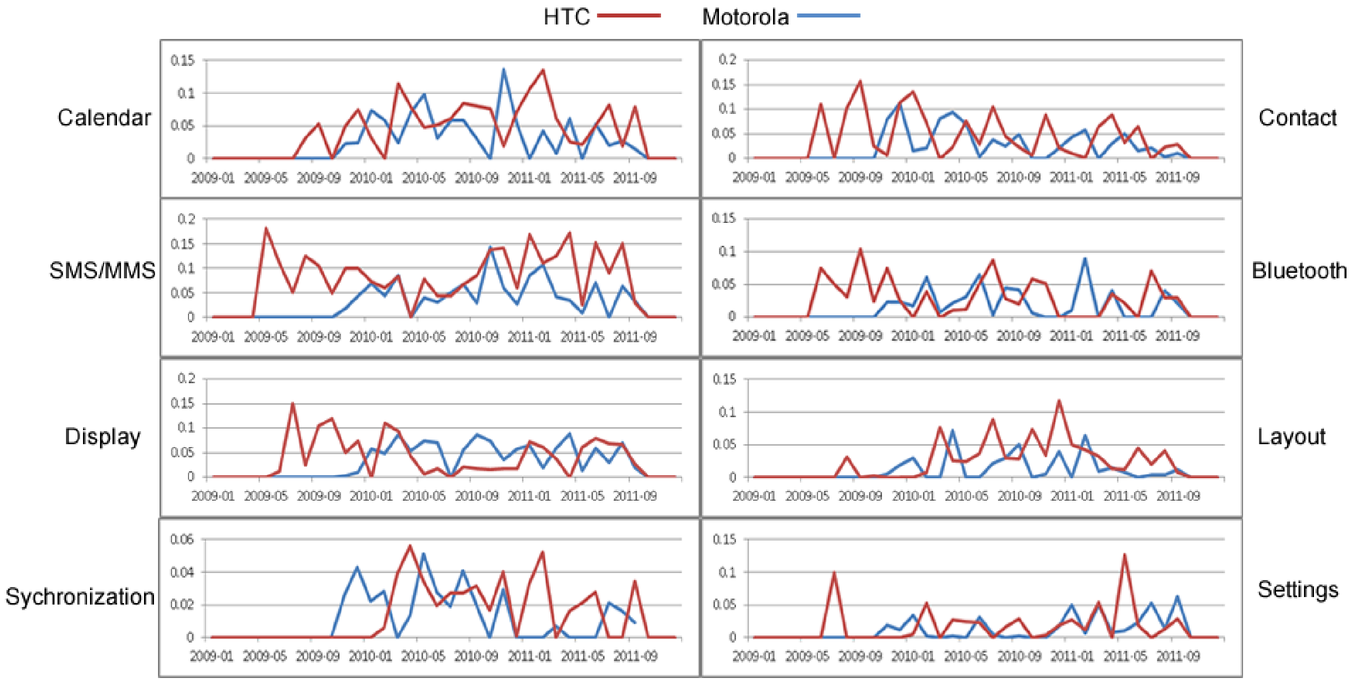

Figure 8. Common Troubled Topics in HTC and Motorola

In order to depict the common topics clearly, we categorize it into two types: Common Troubled Topic and Common Improved Topic.

1) Common Troubled Topic: The Common Troubled Topic means that the relevance of topic has fluctuations all the time for both HTC and Motorola. Figure 8 shows the average relevance of a representative subset of eight common troubled topics for HTC and Motorola over time. The observation that the relevance of the eight topics fluctuates all the time for both vendors indicates that the functions in Android phone for both vendors have problems through.

In fact, each function cannot exist individually, and many functions have to correlate with others. New features introduced are sure to affect other related functions as well. In order to interpret the various topic evolutions and find out the story behind each topic, we here bring up with some cases about correlation among multiple functions and some functions with new technologies.

- SMS/MMS
  Based on the bug reports, problems of SMS in both vendors in the early days are mainly about basic functions, such as “deleting multiple messages causes phone freezes”, “randomly delete all the text”. Afterwards, the upgrade from Android 2.1 to Android 2.2 results in SMS cannot work well any more. In addition, with the upgrade between major version releases or minor version release, some new introduced features related with SMS lead to some problems. For example, the support of synchronization between phone and other platform brought in some bug reports about SMS synchronization [14].
- Contact
  In Android 1.6 [15], a new feature called Quick Search Box, which provides the search function from contact brought in some problems for Contacts. In Android 2.2[15], the new feature that Bluetooth can share contacts with other phone also contributed some problems.
- Calendar
  According to the bug reports, Calendar experienced from the problem of incorrect display and reminders to the problems caused by synchronization between the phone and Google application or other platform.
- Email
  Similar with Calendar and Contact, it underwent the basic function problems and then other problems caused by the combination of synchronization and Android applications.
- Bluetooth
  Bluetooth experienced the upgrade of the Bluetooth techniques itself. When Bluetooth 2.1 was introduced in Android 2.1, lots of the bug reports were opened with Bluetooth, which was consistent to the two spikes in Figure 8. In the meanwhile, a new feature in Android 2.2 [15] that supports voice dialing over Bluetooth, share contacts with other phones, and Bluetooth enabled car and desk docks results in problems of other functions.
- Setting

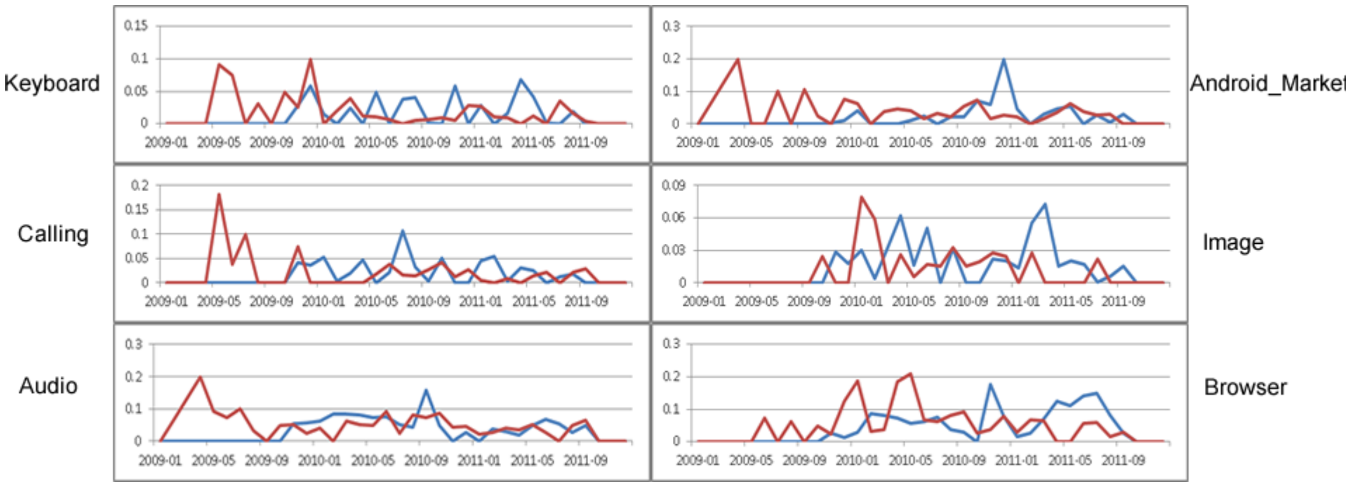

Figure 9. Common Improved Topics in HTC and Motorola

Setting, as the manager of configuration of many functions, it is definitely affected when the new functions are introduced, or the basic features are enhanced.

- Layout
  Similar with Setting, Layout relates with all functions from the user interface level. It is clear in Figure 8 that the relevance is kept higher level over time.

2) Common Improved Topic: The Common Improved Topic means that the relevance of topic turns to be flat after several fluctuations over time for HTC and Motorola. It can also be interpreted as these functions tend to be more robust over time with Android evolution during the whole observed period.

Figure 9 shows the average relevance of a representative subset of 6 common topics for HTC and Motorola over time. Inspecting Figure 9, we can obtain that both vendors have spikes for these six functions.

- The first time of spikes
  The peak relevance for the topics “calling”, “audio” and “android market” arose in 2009, while “browser” and “image” arose in 2010. The observation that the peak values for both vendor occurred in different period stems from the different released time of Android phone from both vendors.
- Number of spikes for each topic
  Topic “keyboard” and “browser” have two peak values. This observation is consistent with the enhancement of Android 1.x to 2.x. In Android 2.0, there is a concrete description about the improvement and enhancement of these two functions[15]. The problem in old version and the adoption of the enhancement in the beginning contribute to these two peaks. The problems in other topics only happened in the early days for HTC, which indicates most problems exist in Android 1.x. Later on, they turned to be flat in Android 2.x.

3) Unique Topic: The Unique Topic means that the relevance of topic has significant difference between HTC and Motorola. It can also be interpreted as these functions cannot work well for specific vendor.

Every vendor has its own branches of Android system [1], even though they base on the same version of Android system. Therefore, it cannot be avoided that there are some specialty issue for different vendors.

Figure 10 shows the average relevance of a representative subset of 2 common topics for HTC over time.

- Language
  Comparing with Motorola who has no bug reports associated with language, Language function should be the specific issue for HTC Android phone. It is implicated by the associated terms of Language topic that bugs were caused by the upgrade from Android 2.1 to Android 2.2 in HTC Desire. In fact, the original Android version in HTC Desire is Android 2.1. The keyboard multiple language function [15] is a new function introduced in Android 2.2. Most of HTC Android phone has no physical keyboard, so this new function has been used frequently by HTC users. In contrast, for Motorola, most of Android phones have the physical keyboard, so this function was used seldom.
- WiFi
  Based on the news report [16], WiFi is a public problem for HTC Android phone. The main problem is the security. In addition, Android 2.2 provides a new feature of the portable WiFi hotspots. This can be the evidence of two spikes for HTC. By Comparison, the relevance of WiFi in Motorola is smooth. This indicates that WiFi in HTC Android phone is a little weaker.

Figure 11 shows the average relevance of a representative subset of two common topics for Motorola over time.

- GPS

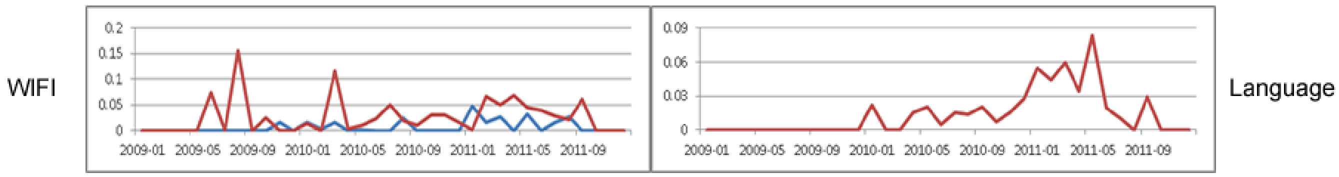

Figure 10. Unique Topics relevance in HTC

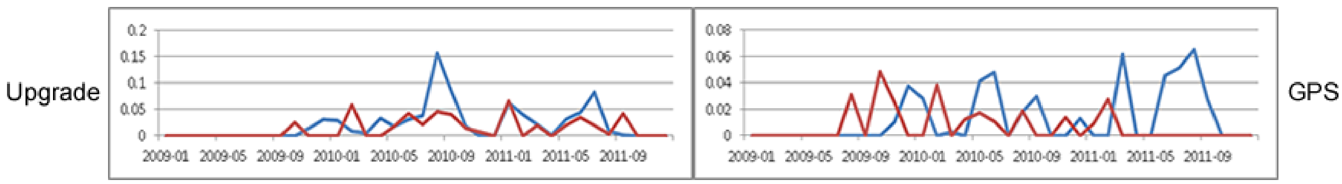

Figure 11. Unique Topics relevance in Motorola

Figure 11 represents that the relevance of GPS in Motorola is higher than that in HTC. Considering the total number of bug reports of HTC and Motorola, it indicates that there are more bug reports associated with GPS topic which makes it as an unique topic. GPS supports in Android system evolved from assisted GPS (A-GPS) to the Google Map. HTC has only one peak relevance; in contrast, Motorola has four spikes. This indicates that GPS support in Motorola branches of Android is not robust.

- Upgrade
  Upgrade has problems for both vendors, it represents two peak values for Motorola. This observation origins from there are mainly two upgrade cases [17], the first one is from Android 1.x to 2.x, the second one is from Android 2.0 to 2.x. Motorola released Android phone based on both Android 1.x and Android 2.x at the same time. Comparing with HTC which only evolves from Android 2.0 to 2.x, it is obvious that Motorola has more update bugs occurred than that in HTC. The various updates make Upgrade as an unique topic for Motorola.

In summary, common troubled topics and common improved topics for both vendors are caused by defects and evolution of functions in Android system. Some unique problems for each vendor show that the branch of Android from each vendor has its own problems. Even though Android system has some defects in each version, if the vendor can make the best of software and hardware during design, they can earn better performance. The typical case is the Keyboard. Virtual keyboard is supported in Android system, but it had problems in the early stage. Most of HTC Android phones adopt the virtual keyboard, while Motorola is prone to physical keyboard. As a result, HTC gets more bugs in the topics which relate with Keyboard, while Motorola avoids the problems. The different choices have already been exposed by the bug reports.

VII. Discussion

Comparing to Apples closed ecosystem for iOS, Android system has many existing fragmentation. These fragmentation include Android systems with five major versions, which are Donut, Eclair, Froyo, Gingerbread, and Ice Cream Sandwich, Andriod system running in devices in various sizes, which from smart phone to TVs, and customized Android system from device vendors [1].

Some people argues that there should be many fragmentation issues since there are too devices running different operating systems [1][2]. On the other hand, there is an opposite viewpoint that there are no fragmentation issues [3].

In this session, we want to discuss this question and try to give a reasonable answer based on our analysis.

A. Does Android have fragmentation issues?

In this paper, the fragmentation of Android system involved in the bug reports are Android 1.x, 2.x and branches of Android systems on multiple devices in HTC and Motorola.

According to the analysis result, common troubled topics existing in both HTC and Motorola Android device indicates that no matter how different the Android system is and how different the device is, the topics turns to be up and down all the time. Common Improved topics of both HTC and Motorola indicate that the problems were caused by the fragmentation of Android system with different versions and that of Android system from branches of both vendors. Unique topics for HTC and Motorola indicate that the bugs were caused by the branches of Android system from vendors.

Besides the three topics, upgrade bugs in both vendors can also be an evidence of fragmentation issues. Before upgrading Android system on devices, each vendor should evolve their own branch of Android system based on the combination of the original version on that device and the

Table III
COMMON TOPICS AND ASSOCIATED WORD LIST WITH RELATED TOP 10 TERMS

Topic Type	Label	HTC	Motorola
Common Troubled Topics	SMS/MMS	message, sms, text, thread, time, conversation, send, version, app, screen	message, text, sms, droid, send, thread, person, threads, number, http
	Email	email , mail, gmail, app, message, send, interface, thread, time, new	email, droid, account, gmail, mail, file, version, open, device, app
	Calendar	calendar, event, day, events, google, reminder, appointment, edit, running	calendar, event, droid, google, appointment, outlook, milestone, data, app, version
	Contact	contact, contacts, number, freed, activity, starting,desire, user, version, field	contact, contacts, droid, number, numbers, behavior, different,list, option, gmail
	Display/Screen	screen, version, desire, behavior, app, home,user, black, new, power	droid, screen, button, correct, home, bar, xoom ,device, user, status
	Bluetooth	bluetooth,headset, car, connect, device, desire, 2.2, work, connects, behavior,2.1	bluetooth, headset, droid, device,connected, connection, 2.2, car,pair, time
	Synchronization	contacts, account, sync, exchange, contact, Gmail, policy, new, list, display	sync, google, account, contacts, device, display, groups, list, droid, milestone
	Setting	Volume, sound, set, pattern, default, change, settings, media, dns, screen	Settings, device, menu, turn, network, behavior, right, wireless, headset, mode
Common Improved Topics	Keyboard	keyboard, input, text, key,number, on-screen, mode, field, landscape,virtual	keyboard, droid, keys,text, press, space, box, open, device,landscape
	Browser	page, text, http, open, server, version, desire, client, 2.1, button	droid, page, web, http, open, xoom, html, behavior, milestone, 3.1
	Audio	music, audio, player, file, play, 2.2,sound, playback,reproduce, mp3	music, droid, player, media, audio, files, volume, play, running, genre
	Calling	number, calls, calling, 2.1, receive, called, button, answer, bluetooth, desire,	droid, calls, number, button, answer, incoming, screen, voice, speaker, 2.2
	Android Market	market, app, google, account, download, update, application, user, apps, paid	market, apps, app, device, application, update, download, purchase, google, milestone
	Image	image, gallery, picture, matrix, photo, camera, pictures, version, 2.2, photo	image, droid, wallpaper, gallery, photo, picture, device, file, select,video
HTC Unique Topics	Language	arabic, desire, letters, characters, translation, rcharacter, read, support, sms, hebrew	NONE
	WiFi	wifi, access, network, connection, connect, router, ssid, desire, http, scan	wifi, xoom, connect, hotspot, turn, connection, ssid, radio, signal, hotspots
Motorola Unique Topics	GPS	gps, data, position, location, maps, google, time,lock, latitude, unit	maps, gps, google, app, droid, location, navigation, map,traffic,update,
	Upgrade	gps, data, position, location, maps, google, time, lock, wrong,tag	update, droid, 2.1,2.2, home, http,longer, settings, performance

latest Android system. The bugs that some functions did not work after upgrade could be eliminate by one piece of maintenance of Android system.

Therefore, we can conclude that Android system has fragmentation issues.

B. Who can benefit from the Topic analysis?

In this paper, the topic analysis suggests the robust and evolution of functions in Android system on devices from different vendors. The analysis result can be used by different people.

For Android system community, the findings enable them to find out which functions have severe problems, which functions have fewer problems, and which functions correlated with each other have more problems. It can also make them to distinguish problems of vendors from others. In addition, the clear idea about the problems of vendors makes community to realize what should be documented and emphasized in corresponding documentations. The concrete and detailed documentation can help reduce the chance of fragmentation issues.

For stakeholders, the findings make them have a whole picture about the function robust of their devices. In the meanwhile, they can also get information about other vendors. With the comparison of the same topics between two vendors, other vendors can easily get some lessons learn which guide them during design and customizing their own system.

For developers, the findings make them know how robust these functions would be. Furthermore, combining correlation among different functions in the code level and the functions trends from bug reports, they can easily find out what functions have potential bugs.

VIII. Threats to validity

Data validity Our data originated from MSR Mining Challenge [12] and the dataset only ranges from January, 2009 to September, 2011. In addition, we extracted the bug report of HTC and Motorola with regular expression, which cannot ensure 100% correctness extraction.

Reliability During labeling the bug reports, every annotator followed the same protocol and used the same labels. However, there were only two annotators working on the labeling work. The annotation could be biased considering the few number of annotators and the limited knowledge we would have.

Analysis validity - The lack of information about the various devices for two vendors in bug reports makes it difficult to analyze the fragmentation of Android system on various devices for vendors. The analysis in this paper mainly focuses on the fragmentation of various Android systems and main fragmentation for each vendor.

IX. Conclusion and Future Work

In this paper we studied the Android bug reports for two Android vendors, HTC and Motorola. We applied Labeled LDA and topic analysis on a corpus of manually tagged bug reports with multiple labels. Our results show that Android system has some fragmentation issues. These findings can be used by Android system community, stakeholders, Android device vendors and developers to make project dashboards, process investigation and feature analysis.

For the future work, we will plan to investigate more vendors in order to reveal vendor specific bug topics and get more concrete fragmentation issues analysis.

References

[1] "The pros and cons of Android fragmentation:," http://mashable.com/2012/01/03/android-fragmentation.

[2] "Google Android: It's time to end the fragmentation:," http://www.zdnet.com/blog/gadgetreviews/google-android-its-time-to-end-the-fragmentation/13279.

[3] "On Android Compatibility:," http://android-developers.blogspot.ca/2010/05/on-android-compatibility.html.

[4] S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein, "Validating the use of topic models for software evolution," in *Proceedings of the 2010 10th IEEE Working Conference on Source Code Analysis and Manipulation*, ser. SCAM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 55–64.

[5] 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.

[6] 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 Engineering - Volume 1*, ser. ICSE '10. New York, NY, USA: ACM, 2010, pp. 95–104.

[7] 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.

[8] 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.

[9] 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.

[10] "Stanford Topic Modeling Toolbox:," Website, http://nlp.stanford.edu/software/tmt/tmt-0.4.

[11] A. Hindle, C. Bird, T. Zimmernann, and N. Nagappan, "Relating requirements to implementation via topic analysis: do topics extracted from requirements make sense to managers and developers."

[12] 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.

[13] "A breif history of Android," http://reviews.cnet.com/8301-19736_7-20016542-251/a-brief-history-of-android-phones.

[14] "Synchronization issue:," http://androidadvices.com/backup-android-phone-contacts-sms-computer.

[15] "Android website:," http://developer.android.com/sdk/index.html.

[16] "HTC WiFi issue:," http://www.linuxfordevices.com/c/a/News/HTC-fixes-WiFi-vulnerability-on-nine-phones.

[17] "Motorola Droid:," http://en.wikipedia.org/wiki/Motorola_Droid.