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
{*dhan3, 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 [**?**].

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 [**?**][**?**]. On the other hand, there is an opposite viewpoint that there is no issues about fragmentation in Android community, only differentiation [**?**]. 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[**?**] of bug reports in HTC and Motorola. Researchers have applied LDA and other topic models to a lot of software projects[**?**][**?**][**?**][**?**]. 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[**?**]. 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)[**?**]. 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.[**?**] 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.[**?**] applied a coherence measurement on topics learned by LDA to model the quality of bug reports. Linstead et al.[**?**] performed LDA to generate traceability links for artifacts in software projects automatically. Topic modeling is also utilized by Thomas et al.[**?**] 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 [**?**].

IV. Methodology

Our methodology is to extract bug reports, assign multiple labels to each of them and then apply Labeled LDA on the labeled data. After that we calculate the average relevance of bug reports to each label over time[**?**] and compare them between two Android vendors, HTC and Motorola. 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.

A. Generating the data

Our first step was to extract the Android bug reports and then find those bug reports relevant to HTC and Motorola We use the Android bug reports provided by the MSR Mining Challenge [**?**] and parse store the bug reports, described as XML data, into a database using SQL Server.

Then we selected bug reports that identified themselves as being relevant to HTC or Motorola through a mention of the words HTC or Motorola in the title text, description text of the bug report. We used regular expressions to extract these relevant bug reports (e.g.,$'\%[0-9a-z]htc[0-9a-z]\%'$ and $'\%[0-9a-z]motorola[0-9a-z]\%'$). We then removed all the declined and duplicate bug reports, leaving us with 1503 HTC bug reports and 1058 Motorola bug reports.

Table I
Manual labels from bug reports of HTC and Motorola.

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

B. Research Features as Potential Labels

In order to investigate fragmentation from a feature-oriented perspective we decided we were going to label the bug reports by relevant features in order to look for feature-relevant bug reports for each manufacturer. To help seed the possible feature-oriented labels we studied summary texts about the Android Operating System [2], popular apps within the Android Market [3]. As well we studied hardware comparisons of the popular Android handsets of of HTC and Motorola [4].

C. Developing Labels

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 to study vendor relevant fragmentation of bug reports.

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

[2]Android Operating System summary: http://en.wikipedia.org/wiki/Android_operating_system (retrieved March, 2012)

[3]Android Market: https://play.google.com/store/apps (retrieved March, 2012)

[4]Android Comparison: http://en.wikipedia.org/wiki/Comparison_of_Android_devices (retrieved March, 2012)

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, screen and keyboard.

To ensure consistency and agreement in labelling the authors executed a training methodology. This methodology also ensured a synchronization of labels. Each author, Zhang and Fan, separately labelled each of these 248 bug reports, with labels inspired by the previous research on Android features. After Zhang and Fan labelled these 248 bug reports seperately, the labels were compared and the authors discussed label agreement and disagreement in order to train themselves to consistently label bug report. To help the comparison each authors labeled data was used as input to STMT's implementation of Labelled-LDA which produced a set of topics. The topics and their relevant bug reports were compared to ensure consistent interpretation of the bug reports and their labels.

D. Labelling the HTC and Motorola Bug Reports

Once the labelling rules were agreed upon each author (Zhang and Fan) seperately labeled HTC and Motorola bug reports, taking over 60 man hours of manual labelling effort. Using the previously stated labelling methodology, labels were created as necessary. For example, the label "calculator" was created in order to label bug reports that occurred later that were relevant to this feature as were several bug reports regarding the correctness of the calculator's results.

1304 HTC and 985 Motorola bug reports were labelled 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.

E. Applying Labeled-LDA

Once the bug reports were labeled we wanted to extract the topics associated with the labels. First we had to preprocess the bug reports in order to apply Labeled-LDA to the labeled bug reports. We convert the title and description of each bug report to lowercase, tokenize and filter the words to remove stop words (words that are less than 3 characters and common English stop words such as "all", "about", "the", "that" and "were"), and then produce word counts/distribution per each HTC bug report and each Motorola bug report.

Separately, we applied Labeled-LDA to these preprocessed HTC bug reports and Motrola bug reports. Labeled-LDA then outputs the topic, a word distribution, associated with our label, as well as a document-topic matrix which links our labels to the documents in the each bug report corpus (HTC and Motorola).

The topic analysis is based on these results. In order to visualize the association of a label (an extracted Labeled-LDA topic) to bug reports over time, we grouped the 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. We generated a distribution of average relevance among three years for each label, showed in Figure2, Figure 3, Figure 4 and Figure 5.

F. Applying LDA

In order to compare the performance between LDA and Labeled-LDA in order to see if Labeled-LDA is worth the effort, we applied LDA to the extracted the same bug reports of HTC and Motorola but without our manual labels. We used the same preprocessing method used on the bug-reports used in the Labeled-LDA analysis.

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 topic. This can 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 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 (Han, Zhang and Fan) evaluated the word distribution of each topic together in each case. We determined if topics were distinct enough based on labeling. Given our previous manual labels that were used by Labeled-LDA we tried to label these LDA topics with those labels. If we repeated too much, or too many labels were clustered around a topic, we considered that choice of n to be unfit. The authors chose the $n = 35$, as the topics generated by LDA were distinct enough 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 [**?**], [**?**], [**?**].

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.

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

In order to determine if LDA would generate the similar results to Labeled-LDA we had to compare the results. Both LDA and Labeled LDA produce matrices of the relationship between bug reports of two vendors and the label or topics. 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.

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 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.

The topic-document matrix often contains quiet 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 label 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 in order to explore the match between 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 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 report of Android and analyzed the result from both quantity and quality views.

We start by exploring the quantity of bug reports for HTC and Motorola. Then we compare and discuss the topic relevance over time for both vendors.

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

A. Overview of bug reports in 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 1 depicts a comparison of the quantity of bugs for HTC and Motorola.

From Figure 1, we can see that 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 models survey[**?**], HTC released the first Android model in October, 2008, while Motorola releases the first model in October, 2009. The first bug reports of both vendors are in order of the first model released by them.

In addition, we can see, in Figure1, the spike for HTC happened in September, 2010, and for Motorola it happened in December, 2009. By reading the bug subjects, we found that the spike of HTC was caused by the fact that many people upgraded their model from Android 2.1 to Android 2.2 at that time, and some functions did not work well after upgrade. For example, users could not send message after the upgrade. The Motorola spike was mainly resulted from the upgrade from Android 2.0 to Android 2.0.1. This suggests that bug activity increases are more relevant to Android version than hardware platform.

B. Topics Analysis of HTC and Motorola

First, it should be mentioned that we extracted 72 topics for HTC and 58 topics for Motorola with Labeled LDA.

By calculating the average of bug distributions associated with each topic, we got the average relevance for each topic by month. Examining the average relevance trends of each topic for the same vendor and comparison between both vendors, we categorize the topics into three types. They are common troubled topics, common improved topics, and unique topics. The Common Troubled Topics mean that the relevance of the topic has fluctuations all the time for both

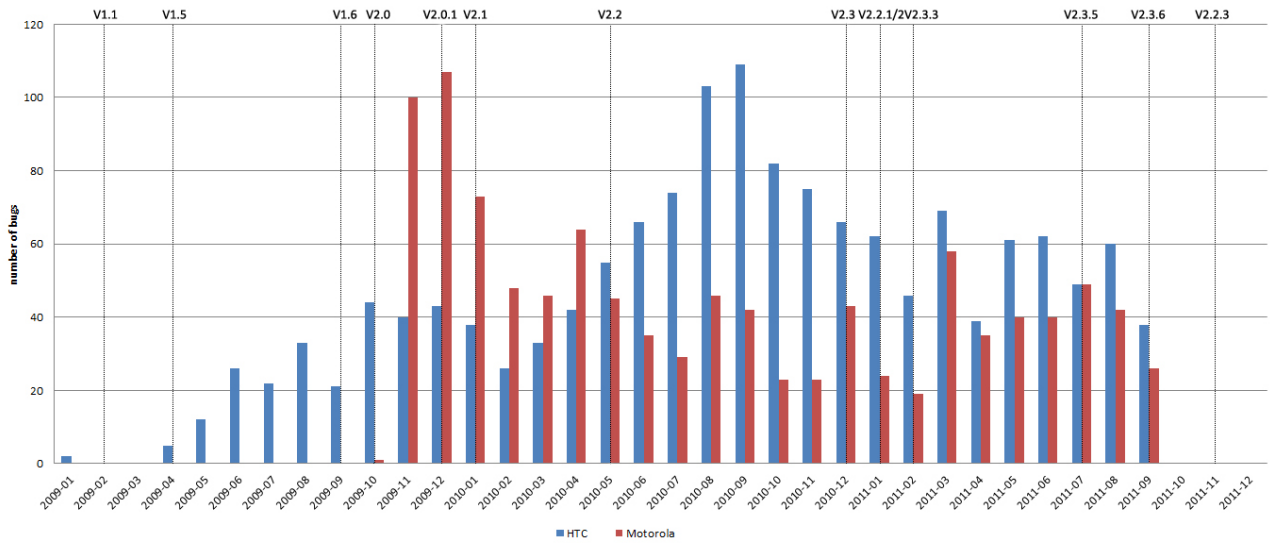

Figure 1. Number of bugs with the major version of Android for HTC and Motorola

HTC and Motorola. The Common Improved Topics mean that the relevance of topic turns to be flat over time after several fluctuations for HTC and Motorola. The Unique Topics mean that the relevance of topic has 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 TableIII, TableIII and TableIII [THE TABLEIII SHOULD BE SPLIETED INTO THREE TABLES]. TableIII shows Common Troubled Topics, TableIII shows Common Improved Topics, and TableIII shows Unique Topics. Each topic is associated with top 15 terms for both HTC and Motorola. These topics are associated with 85% bugs of HTC, and 83% bugs of Motorola. As mentioned before, the label column in TableIII, TableIII and TableIII represents the features of Android.

1) Common Troubled Topic: A representative subset of eight Common Troubled Topics shared by two vendors is shown in Table III and the average relevance of each topic is shown in Figure2.

In TableIII, HTC and Motorola share many identical terms for each label. That means they have the same issues about SMS(*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*), Synchronization (*contact, exchange, google*) and Settings(*turn,network,mode*).

We also found that multiple topics share some same terms for each vendor. For HTC, we can see, 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 model. Calendar and Bluetooth share the same term 2.2 which means Android version 2.2. This indicates that these two topics happened frequently for Android 2.2 in HTC models. For Motorola, seven topics except Settings share the same term droid which means Motorola Droid model. In addition, Calendar and Synchronization in Motorola share milestone which indicates these two topics discussed mostly in Motorola Milestone model. Xoom shared by Display and Settings indicates that Motorola Xoom has more bugs related with these two topics. Furthermore, Synchronization associates with both Xoom and milestone terms. This indicates bugs related with synchronization happened frequently in both Motorola Xoom and Motorola Milestone.

In Figure2, HTC and Motorola share the same topic evolution trends. Both of them have continuous spikes and drops for each topic over time. That indicates bugs associated with these topics have no obvious decreasing trends with Android evolution.

In summary, both of HTC and Motorola have some topics have strong correlation with Android 2.1 and Android 2.2. With Android evolution, these topics evolution do not demonstrate the decreasing trends with Android evolution as we expect. Both of vendors have some topics associated with their typical models. As the topics represent features in Android, we can see there might be a compatibility issue for some Android features.

2) Common Improved Topic: A representative subset of six Common Improved Topics shared by two vendors is shown in Table III and the average relevance of each topic is shown in Figure 3.

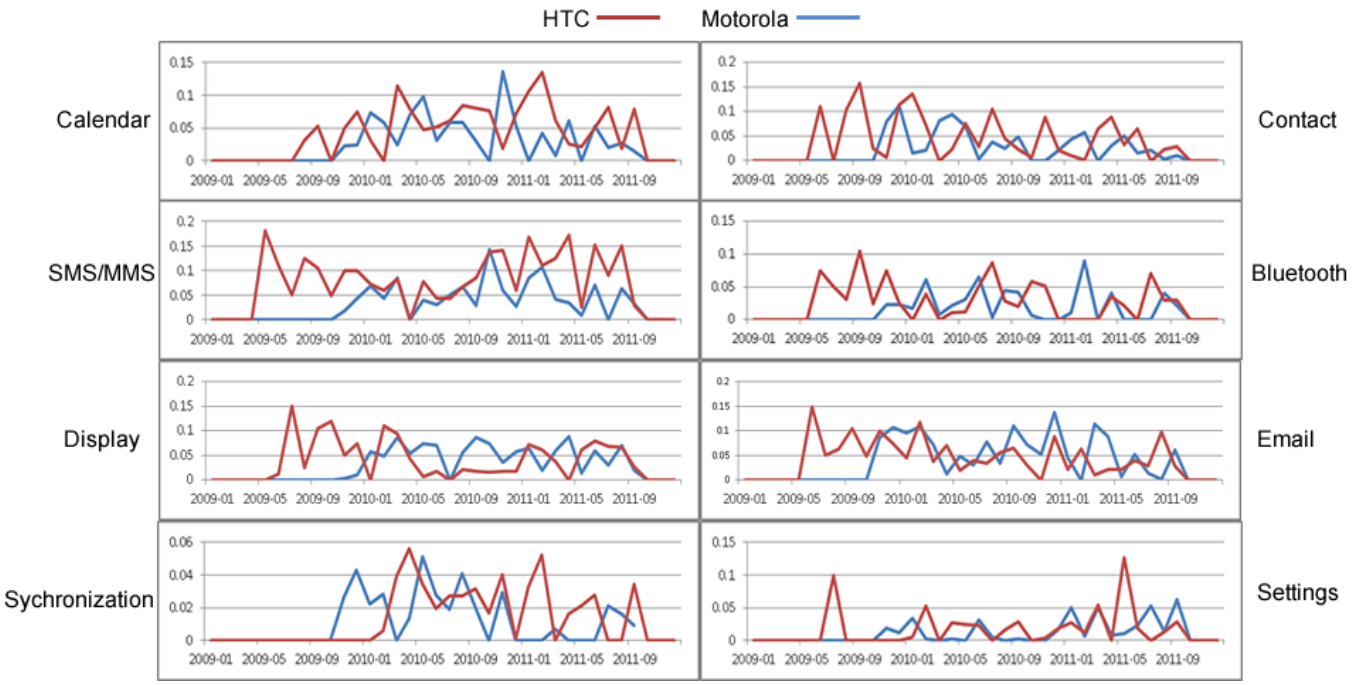

Figure 2. Common Troubled Topics in HTC and Motorola

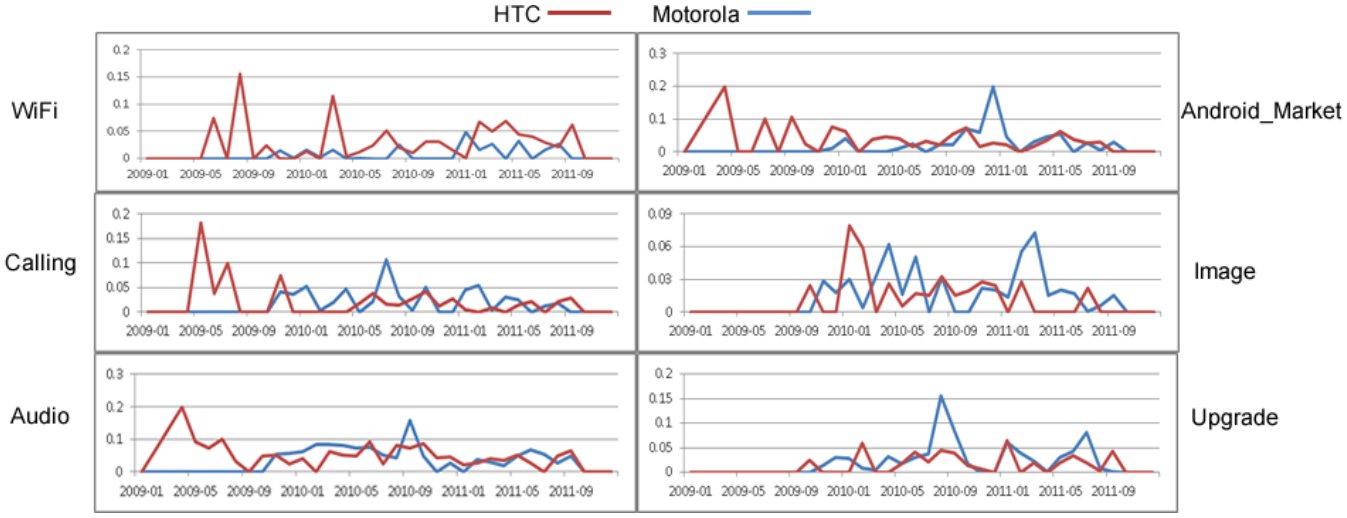

Figure 3. Common Improved Topics in HTC and Motorola

In Table III, HTC and Motorola share many identical terms for WiFi (*connection,ssid,network*), Upgrade (*2.2,2.2.1,http*), Image(*gallery,picture,photo*) and so on. Bugs associated with Upgrade were result from upgrade from Android 2.1 to Android 2.2 in both vendors. This indicates Android 2.2 might have compatibility issue.

In the meanwhile, they also own some special terms. For HTC, bugs related with Calling happened frequently in Android 2.1, and bugs related with Image and Audio happened frequently in Android 2.2. For Motorola, bugs related with Calling happened frequently in Android 2.2 and bugs related with Keyboard happened frequently in Android 2.0.1. The topics also have strong correlation with the Android models for each vendor.

From Figure3, we can see 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, we can see that the same topics from both

vendors have different correlation with Android versions. With Android evolution, these topics evolution do demonstrate the improved trends with Android evolution as we expect. These topics still have strong correlation with their typical model for both vendors.

3) Unique Topics: There are the two unique topics for HTC shown in TableIII. And Figure4 shows the average relevance of each topic.

In TableIII, only HTC has the Language (*arabic, desire, language, 2.2, letters, characters, translation, character, read*) topic. The associated terms indicate that bugs related with language happened frequently in Android 2.2. This stems from the fact that the keyboard multiple language function is a new function introduced in Android 2.2. Moreover, most of HTC models have no physical keyboard, so this new function has been used frequently by HTC users. In contrast, for Motorola, most of models have the physical keyboard, so this function has been used seldom. 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 figure4, HTC keyboard turns to stay in its value, while Motorola has spikes and drops over time. HTC language has the relevance distribution, while there are too few bugs related with language to make language as a topic in Motorola.

There are the two unique topics for Motorola shown in TableIII. And Figure5 shows the average relevance of each topic.

In TableIII, HTC and Motorola, on the one hand, 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*). On the other hand, they have special terms separately. For browser, Motorola has droid, milestone and xoom terms together. This indicates that the browser bugs happened frequently in three Motorola models. This indicates that Browser feature has portability issue within Motorola Android models.

In Figure5, comparing two vendors, we can see the relevance for GPS and Browser demonstrates different trends. For HTC, they have stripes and drops in the early stage, and then stay in their values. For Motorola, they stay in their values and then have stripes and drops afterwards.

In summary, for different vendors, the same topic show significantly different relevance as a result of the different model designs. Within the same vendor, the associated terms implicate that some features have portability issues across models.

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, upgrade topic 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 comparability issue in some features.

From Common Improved Topics and Unique Topics, we can see the same topic from different vendor has different correlation, and they have strong correlation with some specific vendor's models. 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 some features has strong correlation with vendor's model. In the sense of Android model 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 noises 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 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.

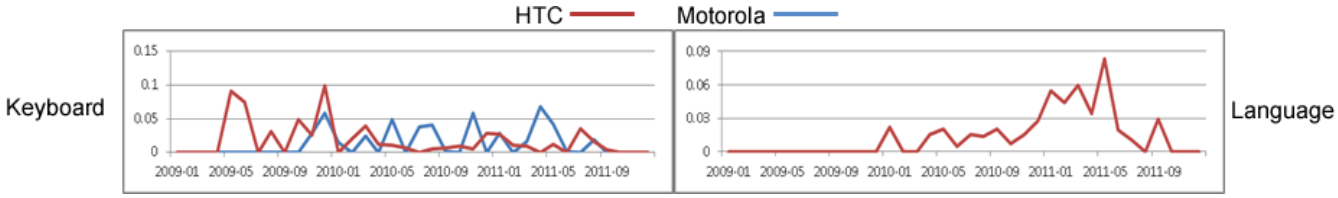

Figure 4. Unique Topics relevance in HTC

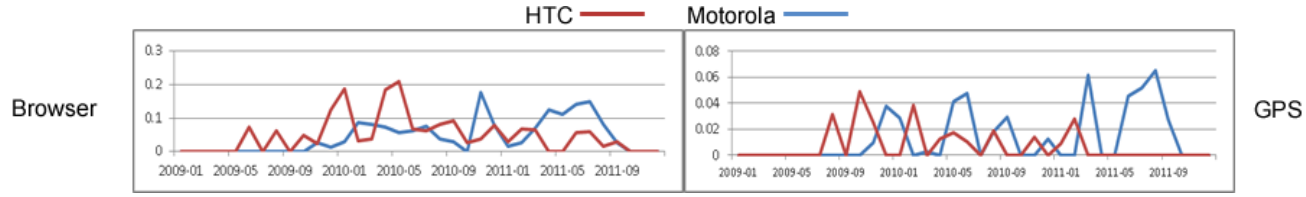

Figure 5. Unique Topics relevance in Motorola

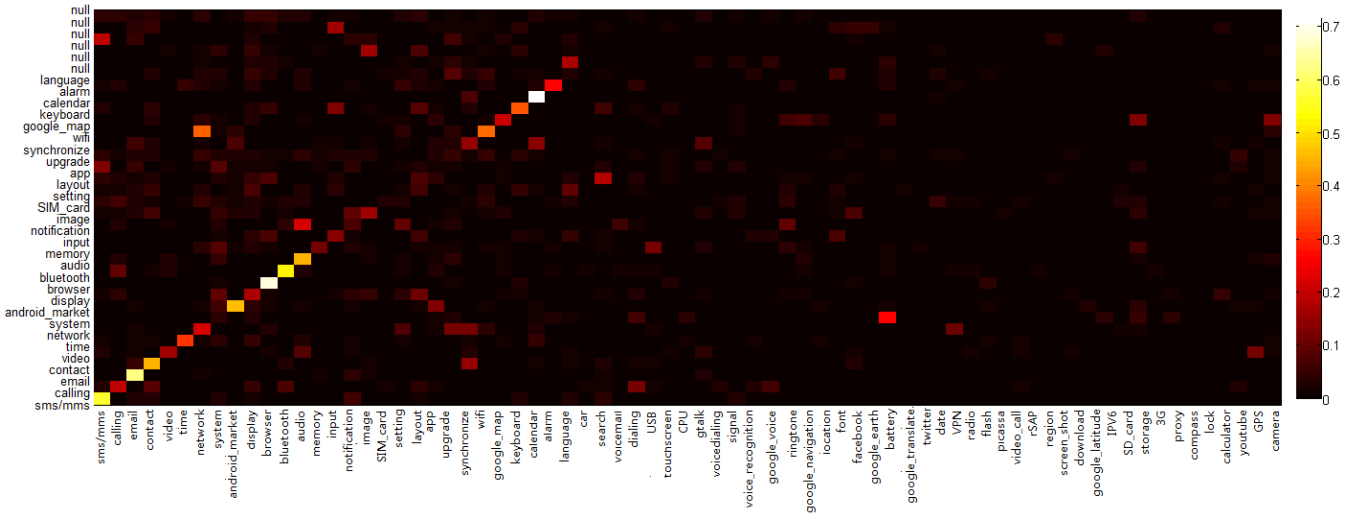

Figure 6. Jaccard similarity of labels between LDA and Labeled-LDA in HTC. X axis is the labels in Labeled LDA and Y axis is the labels of topics generated by LDA. The label "null" in the Y axis means that topic cannot be labeled. The result is based on the HTC bug reports under the threshold of document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

VIII. Threats to Validity

Construct validity Our data originated from MSR Mining Challenge [**?**] 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 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.

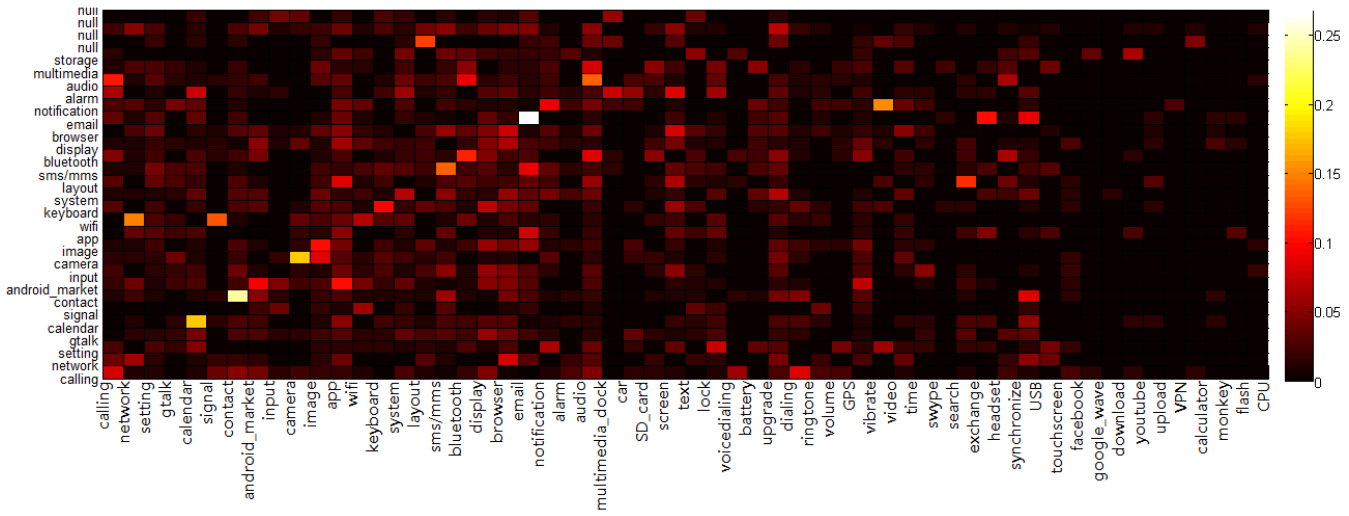

Figure 7. Jaccard similarity of labels between LDA and Labeled-LDA in Motorola. X axis is the labels in Labeled LDA and Y axis is the labels of topics generated by LDA. The label "null" in the Y axis means that topic cannot be labeled. The result is based on the Motorola bug reports under the threshold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

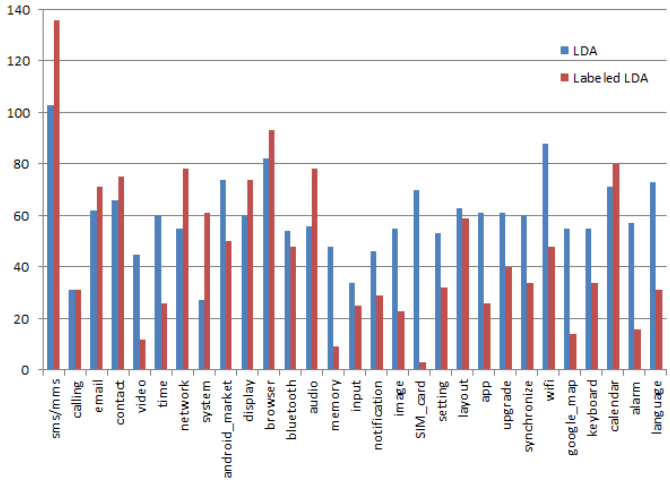

Figure 8. Comparison of number of bug reports related to the same labels from LDA and Labeled LDA in HTC. The X axis is the same labels from LDA and Labeled LDA and the Y axis is the number of bug reports.

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.

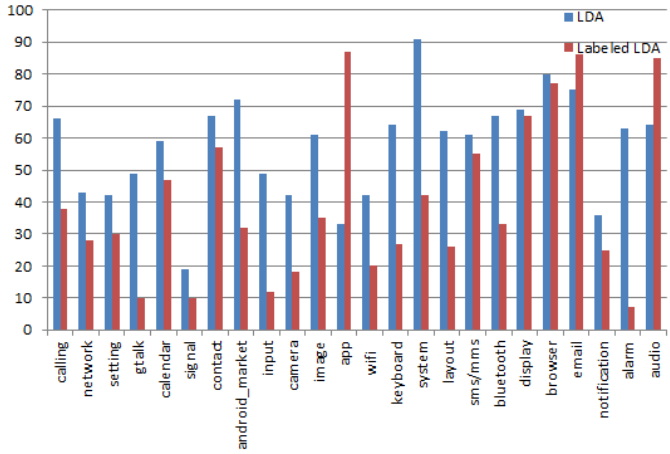

Figure 9. Comparison of number of bug reports related to the same labels from LDA and Labeled LDA in Motorola. The X axis is the same labels from LDA and Labeled LDA and the Y axis is the number of bug reports.

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