Understanding Android Fragmentation with Topic Analysis of Vendor-Specific Bugs

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 causes portability and compatibility issues within the entire Android platform, which increases developer workload, delays application deployment, and ultimately disappoints users. This subject is discussed in the press and in scientific publications but it has yet to be systematically examined. The Android bug reports, as submitted by Android-device users, span across operating-system versions and hardware platforms and can provide interesting evidence about the problem.**

In this paper, we analyze the bug reports related to two popular vendors, HTC and Motorola. First, we manually label the bug reports. Next, we use Labeled-LDA (Latent Dirichlet Allocation) on the labeled data and LDA on the original data, to infer topics. Finally, by examining the relevance of the top 18 bug topics for each vendor's bug reports over time, we classify topics as *common* or *unique* (vendor-specific). The latter category constitutes evidence of fragmentation and lack of portability. By comparing Labeled-LDA against LDA, we find that Labeled-LDA produced better, i.e., more feature oriented, topics than LDA. In this paper we find out how fragmentation is manifested within the Android project and we propose a method for tracking fragmentation using feature analysis on project repositories.

***Index Terms*—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 volatile [1]. Together, iOS and Android constitute almost 86% of the US smartphone market [2]. These two very successful platforms are very different from each other. While Apple tightly controls the software (iOS) and the hardware (iPhone) platform and its evolution, there exists a variety of Android phones produced by different vendors, which often come with custom software, thus giving rise to claims that Android [3] suffers from hardware-based and software-based fragmentation. The term *hardware-based fragmentation* refers to the fact that at any moment, devices based on the same Android operating system run on different processors, graphics cards, and screen sizes [3]. It causes some features in the same Android version to present different issues on the diverse devices. The term *software-based fragmentation* refers to three related aspects. First, there exists parallel deployed versions of the Android operating system. Second, vendors offer customized device-specific Android versions Finally, carriers also offer software customizations.

Fragmentation implies an opportunity for personalization and increased usability, since it enables users to choose the device and software that best meets their needs. Alternatively, it also implies that, due to insufficient cross-platform testing [4], Android applications may not behave consistently across devices and versions of Android. Fragmentation can delay updates for some Android users until their specific Android devices are fully supported. These problems may cause users to lose confidence in the Android platform and damage the brand's market share. Given its potential impact, Android fragmentation is a topic of much discussion and controversy. However, there has been little empirical evidence on if and where the fragmentation specifically exists [5], [6], [7].

Our study focuses 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 October 2008. HTC has made more than 30 different Android phones since then. Motorola produced their first Android phone in October 2009 and has released more than 20 different Android phones since then.

Our objective in this study is to search for evidence of Android fragmentation within the bug reports submitted by users of Android devices. Due to the large volume of Android bug repository, we found that topic models and topic analysis would be convenient to help us extract a set of topics from the repository and then understand it. We first applied topic analysis on the bug reports across two different popular vendors. Several topic-analysis methods have been used by researchers in software engineering, including Latent Dirichlet Allocation (LDA) [8], [9], Latent Semantic Indexing (LSI) [10], and Labeled Latent Dirichlet Allocation (Labeled-LDA) [11]. We applied both Labeled-LDA and LDA topic-analysis to the two sets of vendor-specific bugs and then we compared the two sets of topics we discovered; topics unique to each vendor are concrete evidence of fragmentation. We also compared the results of Labeled-LDA and LDA to see if they produce different results, as each requires a different amount of manual effort to use.

This paper makes three important contributions.

- First, it provides empirical evidence about Android features that contribute to hardware-based fragmentation.

- Second, and equally importantly, it lays out a method, that can be used to examine fragmentation within systems, such as Android, that support numerous hardware platforms.
- Third, by applying both Labeled-LDA and LDA we evaluated if the extra manual effort used to label bug reports for Labeled-LDA is worthwhile.

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

II. Background and Related Work

Topic analysis has been widely used in a variety of software engineering text processing applications. For example, it has been applied to index texts automatically in order to retrieve information, such as similar bug reports [12], [13]. In a nutshell, topic analysis extracts and evaluates the topics from a corpus of text documents through topic models. With topic models, documents can be associated with topics within them, and thus the entire corpus can be indexed and organized in accordance with this discovered semantic structure [12]. In this paper, a topic means a word distribution extracted from bug reports by Labeled-LDA and LDA. A label is the annotation of such a topic.

Latent Dirichlet Allocation (LDA) is a popular probabilistic unsupervised algorithm that models each document as a mixture of topics [11]. LDA automatically learns a set of terms for each topic from a corpus without any constraints [14]. It represents text documents as mixtures of latent topics, where topics correspond to word distributions present in the corpus. Although it is widely used, it suffers from some important shortcomings: it often produces some topics that are hard to interpret, and it is difficult to generate topics that suit a specific purpose. In addition, it requires the number of topics n as an input, but the optimal number of topics can be subjective [15].

Labeled-LDA is an extension of LDA. Labeled-LDA discovers a set of topics by restricting the topic model to use only those topics that correspond to a document's label set [11]. Like LDA, Labeled-LDA models each document as a mixture of underlying topics and generates each word from one topic [11]. Unlike LDA, Labeled-LDA is a supervised algorithm that builds topics using the manual-assigned labels. Therefore, Labeled-LDA can obtain meaningful topics, with words that map well to the labels applied [16].

Topic models have been used to help understand software systems features and link their artifacts together. Marcus et al. [10] used Latent Semantic Indexing (LSI) on both source code and user queries and then identified the most relevant source code documents with similarity measurements. Lukins et al. [13] localized the bugs by retrieving source code with LDA-based static analysis techniques. Asuncion et al. [8] applied a coherence measurement on topics learned by LDA to model the quality of bug reports. Linstead et al. [9] performed LDA to generate traceability links for artifacts in software projects automatically. Grant et al. [15] generated a series of LDA models of source code and estimated the best number of latent topics by using heuristics. Thomas et al. [17] studied the evolution of topics within software projects. Hindle et al. [18] investigated whether the topics extracted by LDA make sense to practitioners. Martie et al. [19] revealed the Android features that are more problematic in a certain period by performing LDA and statistical trend analysis on Android bug reports.

While these studies used LDA to extract topics, we applied both Labeled-LDA and LDA to obtain the topics. In our work, we used the Stanford Topic Modeling Toolbox's (STMT) [16] implementation of Labeled-LDA. We first manually labeled the bug reports with multiple labels and then employed Labeled-LDA to associate topics and documents with the labels we provided [11]. Our technique overcomes some disadvantages of unsupervised algorithms but at the expense of manual labeling.

III. Methodology

Our method for investigating Android fragmentation using topic analysis involved the following steps:

1) First, sets of vendor-specific bug reports are extracted from the Android bug repository.
2) Next, each bug report is manually labeled using feature-oriented terms used by Android developers.
3) Third, we apply LDA to the original bug-report sets and Labeled-LDA to the labeled sets, as produced in step 2.
4) Next, we calculate and visualize the average relevance of each bug report to each topic over time.
5) We then compare the above results between the two vendor-specific sets (HTC and Motorola, in this paper) in order to look for how fragmentation is manifested through an analysis of common and unique topics.
6) Finally, we also compare the performance of LDA topics versus Labeled-LDA topics by comparing the similarity of each pair of topics from LDA and Labeled-LDA.

A. Generating the data

First, we extracted the Android bug reports by parsing and storing the bug reports provided by the MSR Mining Challenge 2012 [20] as a table in a SQL Server database.

Then we selected bug reports relevant to HTC or Motorola if they mentioned HTC or Motorola in their title text or their description text. We then removed all of the declined (unaccepted) and duplicated bug reports, leaving us with 1503 HTC bug reports and 1058 Motorola bug reports.

B. Creating Labels and Training Annotators

To investigate the fragmentation from a feature-oriented perspective we labeled the bug reports by their relevant features. This allows 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

TABLE I
MANUAL LABELS APPLIED TO 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 voicedialing 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 synchronization 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 voicedialing battery upgrade dialing ringtone volume video time swype search exchange headset synchronization facebook google_wave download youtube upload monkey flash VPN touchscreen vibrate CPU system notification text lock GPS calculator USB

descriptions [1] of Android's 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 the Android bug reports. Following the approach of generating labels taken by Hindle et al. [21], authors Zhang and Fan selected a set of 248 HTC bug reports to label separately.

To label a bug report, our annotators (Zhang or Fan) read 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. One bug report could receive multiple labels if it is relevant to multiple identified features. Labels were created as necessary: if a label regarding a feature did not exist, it was created. Our labels shown in Table I consisted of the features, applications, and hardware of Android phones such as SMS/MMS, browser, Wi-Fi , GPS, screens and keyboards.

To ensure consistency and agreement in labeling the authors trained themselves in consistent labeling. Each annotator separately labeled each of these 248 bug reports, with labels based on the previous research on Android features. Upon completion, the annotators compared the labels, discussing label agreement and disagreement in order to train themselves to consistently label the 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. The resulting test topics and their relevant bug reports were compared to ensure that annotators had a consistent interpretation of the bug reports and their labels.

[1] Android 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).

C. Labeling the HTC and Motorola Bug Reports

Once the labeling rules were agreed upon each annotator separately labeled HTC and Motorola bug reports, taking over 60 person-hours of manual labeling effort. New labels were created as necessary: e.g., the label "calculator" was created because later in Android's history there were several bug reports about the correctness of the calculator's results.

As a result, 1304 HTC and 985 Motorola bug reports were labeled with multiple labels, leaving 199 and 73 bug reports that could not 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, with 20 overlapping labels .

D. Applying Labeled-LDA

Once the bug reports were labeled we proceeded 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 we produced word distributions 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 produced the topics (i.e., word distributions), associated with our labels, as well as a document-topic matrix which relates the produced topics to the bug reports from HTC and Motorola. We used the value in the document-topic matrix, which is the probability that a given document discusses a given topic, as the relevance to represent the strength of association between a document and a topic (a larger value is more relevant).

Our topic analysis is based on these results. To visualize the association of topics and bug reports over time, we grouped all the bug reports by month, from 2009 to 2011, based on their opened date for each of the two vendors. In order to eliminate other factors that might impact our analysis, such as the different number of bugs of each vendor, we computed the average relevance values of bug reports to this label in each month [18]. 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 for each label, depicted in Figure 2, and Figure 3.

E. Applying LDA

In order to compare the performance between LDA and Labeled-LDA, 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, incrementing by 5, on the bug reports of HTC. Three of the authors (Han, Zhang and Fan) evaluated the word distribution of each topic together for each value of n. We determined if topics were distinct enough based on manually 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 similar results [17], [21], but Grant et al. [15] attempted to provide a more statistical and less subjective method of determining the number of topics.

We applied the same process to the bug reports of Motorola and we chose the number of topics to be $n = 30$. We set the LDA hyper-parameters α and β to 0.01. As described for the HTC bug reports, we also labeled the 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. That is, we wanted to know if the LDA extracted topics that we manually labeled matched the Labeled-LDA topics that were based on manually labeled bug reports. If the results were similar there would be little point in applying Labeled-LDA in the future, since it takes more manual effort.

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 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,

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

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

where the $\phi(A, d)$ is the set of bug reports that have 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 chose 0.2 as the similarities had the largest mean value. We plotted these pairwise tests (see Figure 4 and Figure 5) 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 (χ^2) on the two sets of distribution to study if each of the two distributions match. The results of comparison are discussed in VI.

IV. TOPIC MINING AND ANALYSIS

In order to investigate fragmentation within Android, we mined the bug reports of Android and extracted topics and analyzed the topics both quantitatively and qualitatively. We started by exploring the distribution of the number of bug reports over time for HTC and Motorola. Then we compared and discussed the distributions of average relevance for each topic over time for both vendors.

A. Overview of Bug Reports in HTC and Motorola

In order to compare the distribution of the number 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 each month for the two vendors. Figure 1 depicts a comparison of the number of bug reports for each HTC and Motorola.

From Figure 1, we can observe that the HTC bug reports were opened in January 2009, and the Motorola bug reports were opened in October 2009. According to the brief history of

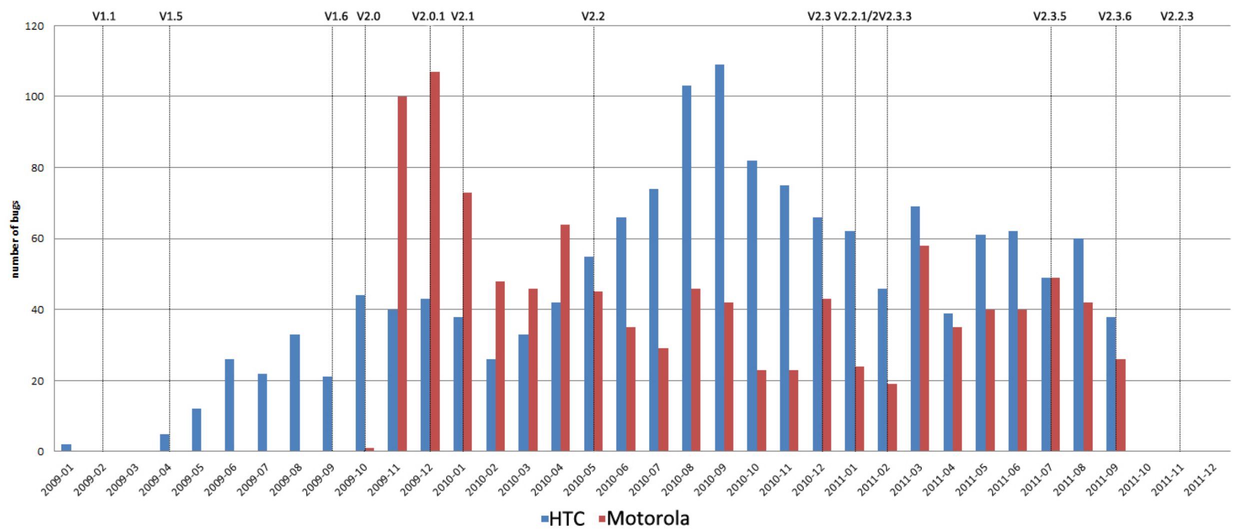

Fig. 1. An overview of the number of bug reports over time for HTC and Motorola. The bottom horizontal axis is the months from January 2009 to December 2011. The vertical dashed lines indicate releases of versions of Android. The vertical axis is the number of bug reports per month.

Android devices survey [22], HTC released the first Android device in October 2008, while Motorola released its first device in October 2009. There is a natural relationship in time between the opened bug reports and the first released Android device of both vendors.

For HTC, out of 109 bug reports shown in the peak of September 2010, 43 bug reports were related with Android 2.1; and 50 bug reports were related with Android 2.2. By reading the bug reports, we found that the peak of HTC was caused by the fact that many people upgraded Android from version 2.1 to version 2.2, and some features did not work well after upgrading, e.g, some users could not send SMS messages anymore. Thus Android upgrades for HTC tend to induce bug reports.

For Motorola, 95 out of 100 bug reports in November 2009 were related to Android 2.0 on the Motorola Droid. Among the 107 bug reports in December 2009, 54 bug reports were associated with Android 2.0 and 53 bug reports were related to Android 2.0.1. We read many of these bug reports to find that most were related to this upgrade. New features were also included in these "upgrade bug reports". Other features that were prevalent in Motorola bug reports were related to Google Maps and the sliding QWERTY keyboard. Much like HTC, Motorola bug reports were often caused by Android upgrades.

B. Topics Analysis of HTC and Motorola

To discuss fragmentation we must consider the issues that each vendor faces that could be due to their own platform. In order to study the similarities and differences of bug reports in Motorola and HTC we used topic analysis to pull out the trends of the bug reports for each vendor. Table I shows the 72 HTC topics and the 57 Motorola topics we obtained with Labeled-LDA. Based on Equation 1, each topic has a distribution of average relevance over time. According to the time-series relevance distribution of each topic, we categorized these topics into *common topics* and *unique topics*. The *common topics* represent the topics that are shared between both vendors, and that tend to share similar distributions of the average relevance over time. The *unique topics* represent those topics with significantly different topic relevance over time (or topics that are completely unique to either vendor).

Table III depicts the top 18 most frequent labeled-topics of HTC and Motorola. Beside each topic we show the top 15 terms generated by Labeled-LDA for each vendor.

1) Common Topics: There are 14 *common topics* shared between the vendors, shown in Table III. Figure 2 depicts common topics between HTC and Motorola. The top 8 topics are busy topics which fluctuate frequently over time, whereas the bottom 6 topics peak and then flatten in interest over time.

The first 8 topics in Figure 2 share many identical topic words between the vendors. Thus the bug reports use similar language between the vendors: 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*), synchronization (*contact, exchange, google*) and settings (*turn,network,mode*). The *Bluetooth* topic has a cross vendor peak with the release of both Android 2.1 and 2.2.

The topics of one vendor tended to share vendor-specific terms. For instance, 9 of HTC's topics, *contact, sms/mms, bluetooth, display, settings, wifi, android market, calling* and *upgrade*, shared the term "desire", which refers to the *HTC Desire* phone. Motorola topics tend to share the term "droid" and the term "xoom", which refers to their *Motorola Droid* and *Motorola Xoom* lines of devices. Motorola topics that mentioned "xoom" included *display, settings* and *synchronize*.

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

Label	HTC	Motorola
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
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
bluetooth	bluetooth, headset, car, connect, device, connection, version, data, app, desire, 2.2, work, connects, behavior, 2.1	bluetooth, headset, droid, device, connected, connect, devices, calls, car, issue, connection, 2.2, car, pair, time
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
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
synchronization	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
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
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, reporoduce, download, purchase, google, milestone
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
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
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
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
keyboard [HTC]	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
language [HTC]	arabic, desire, language, 2.2, letters, character, translation, character, read, support, sms, write, hebrew, devices, 2.3	NONE
browser [Motorola]	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
GPS [Motorola]	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

Thus there is evidence that different product lines faced different issues.

For Motorola, vendor-specific brand names tended to occur in the top topic words of their common topics. Six topics in Figure 2 share many identical terms for wifi (*connection, ssid, network*), upgrade (*2.2, 2.1, http*), and image (*gallery, picture, photo*) in HTC and Motorola. For example, bug reports related with *upgrade* happened frequently in both vendors when people upgraded Android from 2.1 to 2.2. It indicates that Android 2.2 might have some incompatibility issues while upgrading on certain devices.

In summary, both vendors share some similar topics and similar terms associated with these topics and also exhibit similar topics evolution over time. HTC and Motorola topics tend to differ in terms of the product-lines that appear in the topic words. The *HTC Desire*, *Motorola Droid*, and *Motorola Xoom* are often mentioned. Both vendors share some *common topics*, but even within these topics and vendors it seems certain product-lines are affected by different bugs. Thus this is evidence that there are portability issues and compatibility issues relevant to the shared *common topics* of vendors, and even across different vendor's smartphone product lines (*Motorola Xoom* and *Droid* correlated with different topics).

2) Unique Topics: Some topics are more specific to one vendor than the other. In Table III we present 2 unique HTC topics and 2 unique Motorola topics. Figure 3 shows the distribution of the average relevance of each of these unique topics for HTC and Motorola.

Topic *language* (*arabic, desire, language, 2.2, letters, characters, translation, character, read*) is an unique topic of HTC. The associated terms indicate that bug reports related with *language*, and internationalization occurred frequently in Android 2.2. This is because the feature of "multiple keyboard languages" was a new feature in Android 2.2. With this feature, multi-lingual users can add multiple languages to the keyboard and switch between multiple input languages [23]. Most HTC devices have no physical keyboards, so this new feature is frequently used by HTC users. In contrast, Motorola's Android devices tend to have physical keyboards, which might explain the lack of bug activities in the Motorola bug reports. Figure 3 shows that HTC *keyboard* relevance peaks and drops out, while *keyboard* in Motorola is steady. This behavior suggests that hardware and software configuration dictate the importance of the *keyboard* topic. We did not notice internationalization or

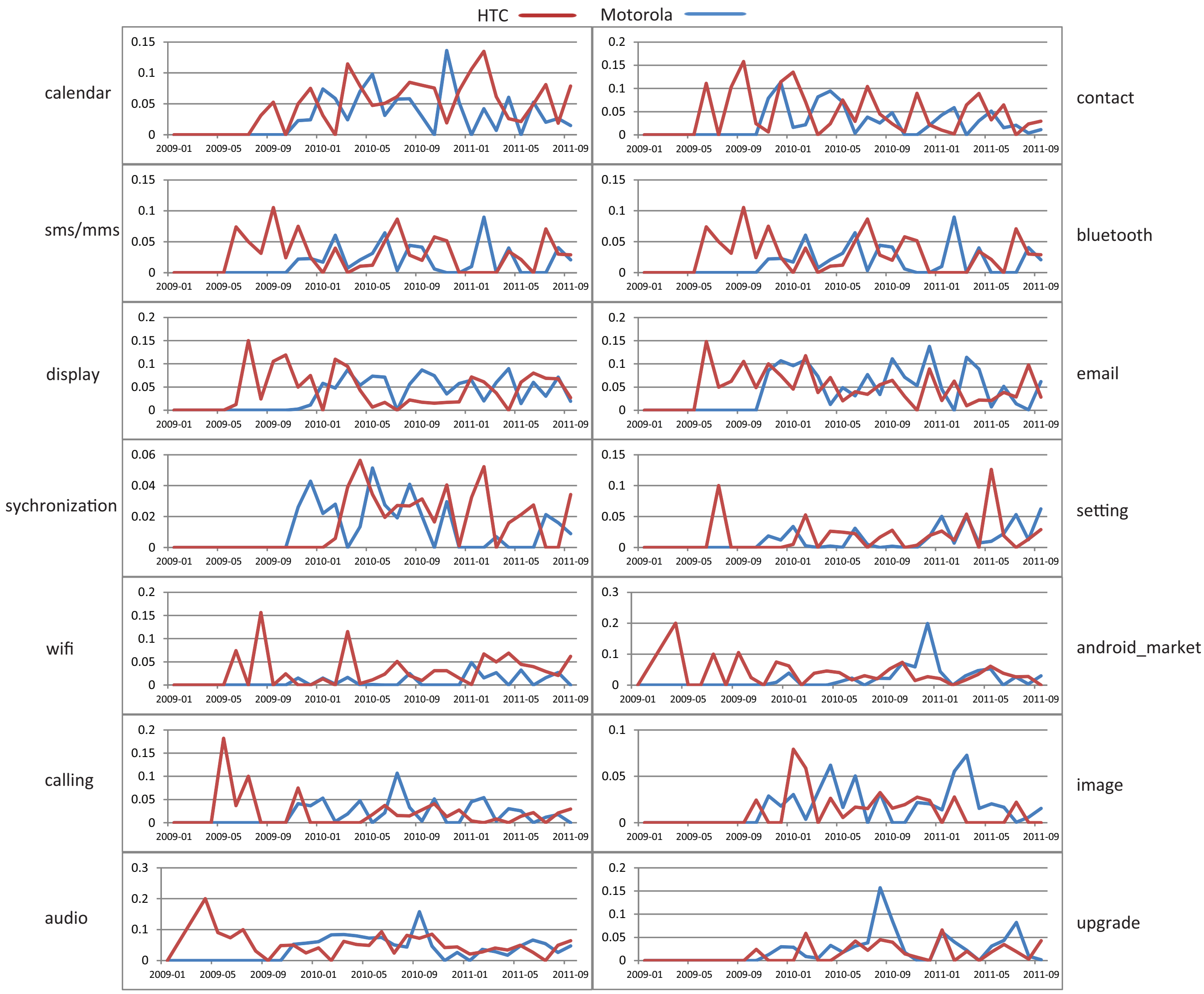

Fig. 2. Relevance of common topics in HTC and Motorola. X axis is months from Jan. 2009 to Sep. 2011. Y axis is the average relevance of topics.

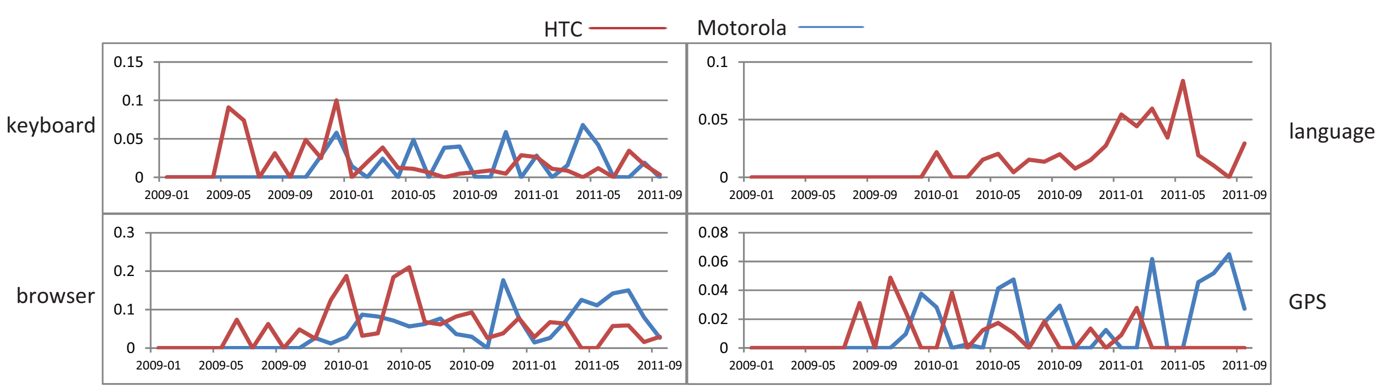

Fig. 3. Relevance of unique topics in HTC (top) and Motorola (bottom). X axis is months from Jan. 2009 to Sep. 2011. Y axis is the average relevance of topics.

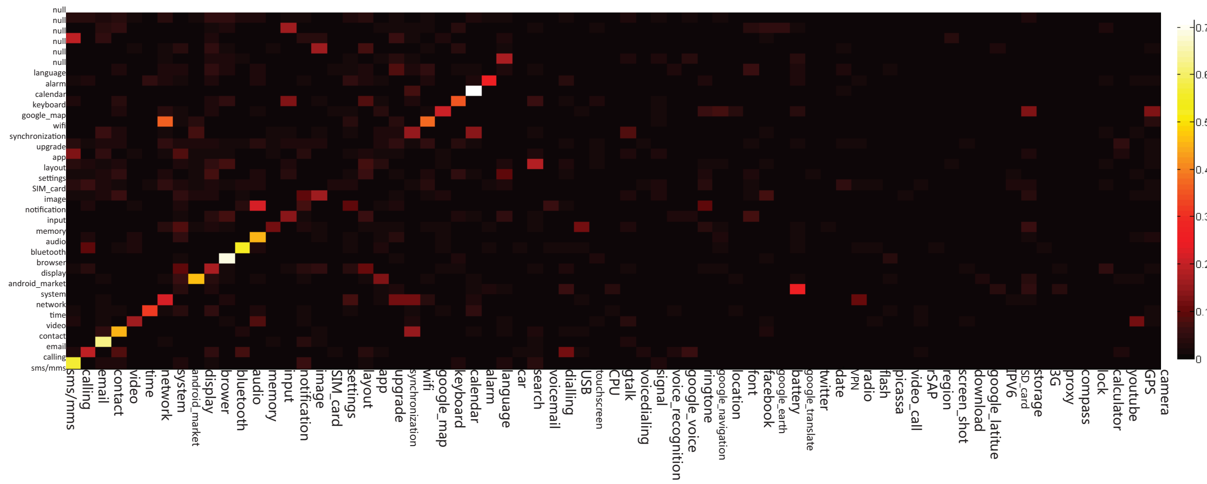

Fig. 4. Jaccard similarity of labels of Labeled-LDA and topics of LDA in HTC. X axis is the labels in Labeled-LDA and Y axis is the labels of topics generated by LDA. The label “null” in the Y axis means that topic cannot be labeled. The result is based on the HTC bug reports under the threshold of document relevance of 0.2. Brighter means higher Jaccard similarity.

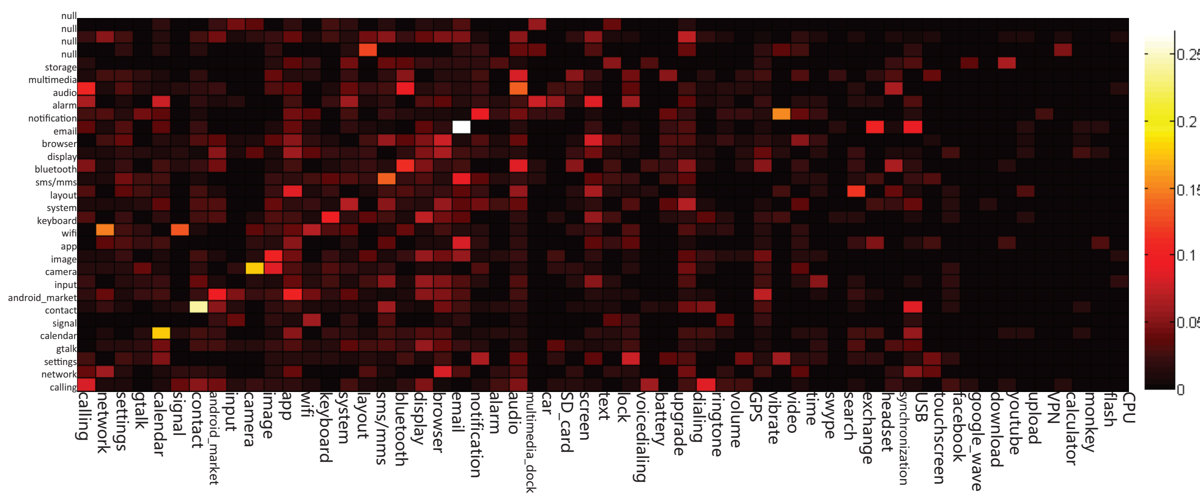

Fig. 5. Jaccard similarity of labels of Labeled-LDA and topics of LDA in Motorola. X axis is the labels in Labeled-LDA and Y axis is the labels of topics generated by LDA. The label “null” in the Y axis means that topic cannot be labeled. The result is based on the Motorola bug reports under the threshold of document relevance of 0.2. Brighter means higher Jaccard similarity.

language issues in the Motorola bug reports that we labeled, thus this issue seemed more HTC specific.

Browser (*browser, page, text, http, open, server*) is an unique topic of Motorola. Table III shows that Motorola's product-lines, *droid*, *milestone* and *xoom*, are relevant to *browser*. Motorola's *GPS* topic (*gps, data, position, location, maps, google, time, lock, wrong, icon, turn, home, latitude*), shown in Figure 3, starts slowly, peaks and drops off while HTC's GPS issues occur much earlier and tend to fall off. Motorola and HTC did not share the same GPS software at this time. Thus *GPS* and *browser* are two topics where both vendors differ.

In summary, for both vendors, they have vendor-specific topics which imply there may be portability issues, and thus give rise to issues of fragmentation, especially in terms of internationalization, keyboards and GPS support.

V. Fragmentation Discussion

In this section, we discuss the Android fragmentation based on the analysis about comparison of the distribution of the number of bug reports, *common topics* and *unique topics* from HTC and Motorola.

Many of our topics, bug reports, and peaks in activity are correlated to Android releases such as versions 2.0, 2.1, and 2.2. It is intuitive that new features and changes to the OS would induce bug reports due to the difficulty in testing across all of these vendors and product-lines. The *common* *topic upgrade*, which share *2.2, 2.1* and *http* in HTC and Motorola, indicates that Android 2.2 has incompatibility issues while upgrading on certain devices. The *unique topic language* in HTC is related to the new feature, “multiple keyboard languages” introduced in Android 2.2. This kind of evidence indicates that there are unique features that cause users to face defects during an Android OS version update. As we analyzed in section IV-B1, the product-lines of HTC (*HTC Desire*) and Motorola (*Motorola Droid* and *Xoom*) share 9 topics in common and correlate with 5 different topics. Whether it is multiple versions of Android in the field or the new releases of Android, we witnessed both different and similar behavior within the bug reports of each vendor and their product-lines. The variation in topic words in our *common topics* tended to relate to the distinct product lines of the vendors. Different product lines were associated with different topics indicating there might be fragmentation issues internal within the product lines of a vendor. The variation of hardware devices in each vendor contribute to the bug topics about both Android features and components of handsets. For example, the topic *display* in Motorola shown in Table III correlates with “droid” and “xoom” which infer to their *Motorola Droid* and *Motorola Xoom* lines of devices. The *unique topics* provide more evidence that this might be the case. Thus, hardware-based fragmentation within Android appears through the portability issues discussed in the unique topics.

When we refer to Android, we mean all the deployed Android versions both from the community, vendors and carriers. We can see that Android has a software-based fragmentation issue and it is evident in the issues when updating, but we lacked the necessary data to talk about the conflicts caused by multiple supported versions of the same operating system. Yet we can tell that the different hardware configurations, especially keyboards, puts a different emphasis on relevant software topics, such as software keyboards, within the bug reports relevant to each vendor.

> Hardware-based fragmentation in Android is evident by differing bug topics and product specific issues.

VI. Comparing of LDA and Labeled-LDA

In this section we investigate if LDA and Labeled-LDA would generate the similar results. This is an important issue because Labeled-LDA took around 60 times the amount of time it took to label the topics extracted by plain LDA (and our data is freely available [24]).

Figure 4 and Figure 5 depict the pairwise Jaccard similarities of labels from LDA and Labeled-LDA. The brighter entries mean the pair of labels have higher Jaccard similarity. These two labels in LDA and Labeled-LDA are similar if they share similar bug reports. The darker entries mean the pair of labels have lower Jaccard similarity and share less bug reports in common.

From these two Jaccard similarity plots (Figure 4 and Figure 5) of topics and labeled-topics 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.

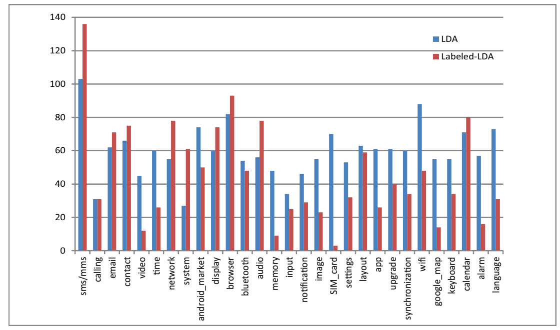

Fig. 6. Comparison of number of bug reports related to the same labels from LDA and Labeled-LDA in HTC. The X axis is the same labels from LDA and Labeled-LDA and the Y axis is the number of bug reports.

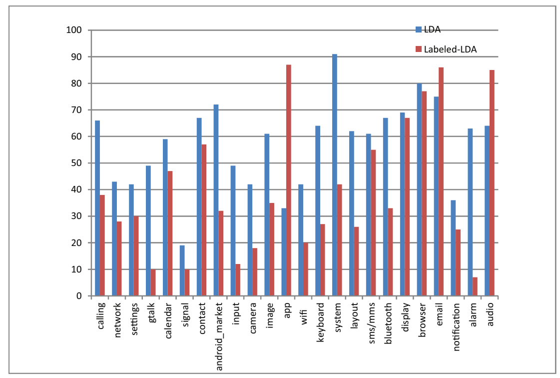

Fig. 7. Comparison of number of bug reports related to the same labels from LDA and Labeled-LDA in Motorola. The X axis is the same labels from LDA and Labeled-LDA and the Y axis is the number of bug reports.

This observation is expected since most of the diagonal entries are the Jaccard similarities between the same labels from LDA and Labeled-LDA. However, even the mean similarities of the diagonal entries 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. Thus the diagonals tend to match, but the Jaccard similarity is lower than we expected.

We inspected the distributions of bug reports associated with topics and labels more closely and provided summaries in Figures 6 and 7. Figure 6 shows the number of bug reports that are related to the same labels in the bug reports of HTC and Figure 7 illustrates the number of bug reports that related to the same labels in the bug reports of Motorola. The number of bug reports related to same labels in LDA and Labeled-LDA are different, this is confirmed by the X^2 tests ($p < 0.01$).

For LDA and Labeled-LDA topics with the same label, we found that LDA predicted fewer of the relevant bug reports and that the relationships between topics and bugs for LDA and Labeled-LDA was often different. We found the Labeled-LDA topics to be of better quality and matched better to our understanding; but we found that Labeled-LDA required up to 60 times the effort that labeling LDA extracted topics required.

> Labeled-LDA produced more feature relevant topics than LDA, but required almost 60 times more effort.

VII. Threats to Validity

Construct validity: Our authors are not Android domain experts. They annotated thousands of bug reports based on their Android expertise. Their own biases could have caused us to measure their biases rather than the features relevant to the bug reports. Also our automatic selection of vendor-specific bug reports might not accurately reflect the vendor-specific issues within the bug repository. Our argument about fragmentation relies on the divergence of topics between vendors. Bug reports are only one source of fragmentation evidence, other repositories might have other kinds of evidence.

Internal validity: We argue that the divergence of topics in terms of relevance and keywords were indicators of fragmentation, thus internal validity could be threatened by choices of parameters and labels.

External validity: This study focused on one project, Android, and only two vendors, HTC and Motorola, thus external validity could be greatly improved by investigating other systems such as FreeBSD that face similar portability and fragmentation issues.

Reliability: Reliability is threatened by the judgment of the two authors used to label the bug reports. We bolster reliability by describing our method in detail and training authors in labeling bug reports.

VIII. Conclusions

In this study we found how fragmentation is manifested within Android by comparing and contrasting the bug topics, extracted from Android bug reports, of two Android smartphone vendors: HTC and Motorola. Based on Labeled-LDA topic analysis we found that even for shared *common topics* there was a divergence in topic keywords between vendors. We found that for each vendor, different topics tended to be associated with their own different products, providing even more evidence of vendor-specific fragmentation. Thus our topic analysis provides evidence of hardware-based fragmentation affecting the bugs reported in the Android bug repository.

We manually labeled 1000s of individual bug reports so that we could apply Labeled-LDA and extract feature-specific topics. We used our labeled bug reports to compare Labeled-LDA and LDA, as LDA is unsupervised and requires far less effort to run than Labeled-LDA. We found that LDA (with manually labeled topics) and Labeled-LDA produced some similar topics. The labeled-topics of Labeled-LDA were more feature specific and more useful to our analysis, yet the cost of labeling bug-reports versus labeling LDA topics is almost 2 orders of magnitude greater in terms of person-hours.

Our findings can be used to make project dashboards, process mining and software process recovery; our method for investigating fragmentation could be applied to other projects, such as Ubuntu or FreeBSD, that suffer from multiple deployed versions and platforms.

References

[1] H. Dediu, "The Global Smartphone Market Landscape," 2011, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/V7s5w

[2] J. Fingas, "Nielsen has Android near 52 Percent of US Smartphone Share in Q2, iPhone Ekes out Gains," 2012, [Accessed 24-August-2012]. [Online]. Available: http://goo.gl/JLgyf

[3] S. Singh, "An Analysis of Android Fragmentation," 2012, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/gNDHC

[4] T. Hooper, "Stop Whining about Android Fragmentation," 2012, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/pJr6C

[5] C. Delessio, "The Pros and Cons of Android Fragmentation," 2012, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/4Uaal

[6] Z. Epstein, "Android Fragmentation Rears Its Ugly Head Once Again," 2012, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/7kOhS

[7] V. Madhav, "Fragmentation in Android: Boon or Bane," 2010, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/zJAZg

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] H. Li and K. Yamanishi, "Topic Analysis Using a Finite Mixture Model," *Information processing & management*, vol. 39, no. 4, pp. 521–541, 2003.

[13] S. Lukins, N. Kraft, and L. Etzkorn, "Source Code Retrieval for Bug Localization Using Latent Dirichlet Allocation," in *Reverse Engineering, 2008. WCRE'08. 15th Working Conference on*. IEEE, 2008, pp. 155–164.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[15] S. Grant and J. R. Cordy, "Estimating the Optimal Number of Latent Concepts in Source Code Analysis," 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. 65–74.

[16] "Stanford Topic Modeling Toolbox," [Accessed 11-July-2012]. [Online]. Available: http://nlp.stanford.edu/software/tmt/tmt-0.4

[17] 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.

[18] A. Hindle, C. Bird, T. Zimmermann, and N. Nagappan, "Relating Requirements to Implementation via Topic Analysis," in *Proceedings of the 2012 International Conference on Software Maintenance*, September 2012.

[19] L. Martie, V. K. Palepu, H. Sajnani, and C. Lopes, "Trendy Bugs: Topic Trends in the Android Bug Reports," in *the 9th Working Conference on Mining Software Repositories*, ser. MSR '12, 2012, pp. 120–123.

[20] 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.

[21] 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.

[22] K. German, "A Breif History of Android Phones," 2011, [Accessed 11-July-2012]. [Online]. Available: http://goo.gl/GXO4D

[23] "Android Developer Website," 2012, [Accessed 11-July-2012]. [Online]. Available: http://developer.android.com/sdk/index.html

[24] D. Han, C. Zhang, X. Fan, A. Hindle, K. Wong, and E. Stroulia, "Annotated Topic Data Used in this Study," http://softwareprocess.es/static/Fragmentation.html, 2012.