Mining Questions Asked by Web Developers

Kartik Bajaj Karthik Pattabiraman Ali Mesbah
Electrical and Computer Engineering
University of British Columbia
Vancouver, BC, Canada
{kbajaj, karthikp, amesbah}@ece.ubc.ca

ABSTRACT

Modern web applications consist of a significant amount of client-side code, written in JavaScript, HTML, and CSS. In this paper, we present a study of common challenges and misconceptions among web developers, by mining related questions asked on Stack Overflow. We use unsupervised learning to categorize the mined questions and define a ranking algorithm to rank all the Stack Overflow questions based on their importance. We analyze the top 50 questions qualitatively. The results indicate that (1) the overall share of web development related discussions is increasing among developers, (2) browser related discussions are prevalent; however, this share is decreasing with time, (3) form validation and other DOM related discussions have been discussed consistently over time, (4) web related discussions are becoming more prevalent in mobile development, and (5) developers face implementation issues with new HTML5 features such as Canvas. We examine the implications of the results on the development, research, and standardization communities.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement

General Terms

Measurement

Keywords

Text Mining, Stack Overflow, Topic Modeling, Web Developers

1. INTRODUCTION

Modern interactive web applications require the integration of many languages on the client-side, such as JavaScript, CSS and HTML. Web developers[1] use HTML to define the initial Document Object Model (DOM) layout, CSS to provide styling to the layout, and JavaScript to interact with that layout. JavaScript is often responsible for the core functionality of a web application, yet it is difficult to program in due to features such as loose typing, dynamic code generation using *eval*, and frequent interaction with the DOM. As a result, JavaScript code often experiences errors [14], which can affect the operation of the web application. Further, CSS code is often ad-hoc and difficult to maintain, which can lead to unnecessary code bloat [12]. Finally, with the advent of HTML5 [8], many new features have been added to HTML, making it potentially error prone and difficult to use. Therefore, to be able to help developers effectively, there is a compelling need to understand programming challenges faced by web application developers.

In this study, our goal is to bridge the knowledge gap between the developer and research communities, and help in developing tools that will increase the overall quality of web application development. To pursue our goal, we conduct a quantitative as well as qualitative study of more than 500,000 Stack Overflow questions related to web development. Stack Overflow[2] is a question and answer (QA) site for programmers. Being one of the active QA sites [24], the data available in Stack Overflow is huge, and can provide high-level insights into the issues faced by programmers in web development. We chose Stack Overflow for our study as the questions contain detailed information about the issues faced by developers, often followed by a discussion and an 'accepted' answer. Our study pertains to QA items related to JavaScript, HTML5, and CSS, on Stack Overflow.

We are not the first to use Stack Overflow for understanding issues faced by programmers. Prior work has used Topic Modelling on Stack Overflow questions to list the categories of discussions [4] [2] [17], or used Stack Overflow statistics to analyze user behaviour [20] [16] [3]. However, none of these papers examine fine-grained aspects of Stack Overflow data related to web applications' development. Doing so requires new kinds of heuristics to analyze the data, and to gather insights from it. To the best of our knowledge, we are the first to analyze Stack Overflow data with regard to modern web application development, and to extract actionable insights from the data for the developer and researcher communities.

Our work makes the following main contributions:

- We define novel heuristics for analyzing Stack Overflow questions based on participating user reputation; this is needed since existing ways of ranking reputation do not satisfy our criteria for extracting important information (Section 3.1);
- We categorize related discussions into multiple categories based on the dominant topics in the interactions among developers. We then highlight the important topics of discussion from these categories.

[1] In this paper, when we say web development, we mean client-side web development, unless we say otherwise.

[2] http://stackoverflow.com

MSR '14, May 31 – June 1, 2014, Hyderabad, India

- We identify temporal trends in related discussions to understand current and future trends in the area; and
- Finally, we devise a metric to rank Stack Overflow questions based on the contributions by registered users and qualitatively analyze the top 50 questions.

The main findings from our study are (1) cross-browser related discussions while prevalent in the past, are becoming less important, (2) DOM APIs and event handling issues have been a significant source of confusion for web development, (3) HTML5 is gaining popularity in (mobile) web applications, (4) web related topics are becoming more prevalent in mobile development, though the topics are broadly similar to those in other web applications, and (5) even expert programmers are confused by some of the new features added to HTML5 and JavaScript.

2. BACKGROUND AND MOTIVATION

This section provides background information about modern web applications, followed by a brief description about Stack Overflow and its data dumps. Finally we describe the goal and motivation of our study.

2.1 Web Applications

Modern web applications consist of both client and server side components. In this paper we focus on the client-side of the web applications, which consist of the following aspects:

JavaScript is a prototype-based scripting language with first-class functions. JavaScript is mainly used to (1) attach various events to the DOM tree, (2) dynamically change the state of DOM tree by modifying the elements or their attributes by calling DOM API access methods, and (3) communicate asynchronously with the server. JavaScript is event-based, dynamically typed, and asynchronous in nature. While JavaScript is predominantly used in the client-side of web applications, it is becoming increasingly popular in server-side applications, game engines, and desktop applications.

HTML5 HTML is used to define the layout of the web page. Internally, the browser generates a Document Object Model (DOM), which is a hierarchical representation of the state of elements in the web page. Changes in the value of any of these elements are reflected on the rendered page. HTML5 is the latest version of HTML and it marks a significant improvement over the previous versions. The main goal of HTML5 is to increase the human readability of the code and include native support for multimedia features such as audio and video. HTML5 has added new HTML tags such as `canvas`, and has introduced new attributes for the existing tags to provide additional information in a systematic manner. New JavaScript API's have also been introduced as part of HTML5 specification.

CSS is a design language used to define the presentation of the web document. It can be used to modify style properties and change the presentation of a particular node or a group of nodes in the DOM tree.

Mobile application development has also been largely influenced by the advancement in HTML5 and CSS3 [23]. HTML5 is becoming a common platform for mobile application development, and companies are investing significant resources in supporting it [1].

2.2 Stack Overflow Dataset

Stack Overflow is a popular community-driven questioning and answering service. It has been actively used by programmers to ask questions [24]; from January 2009 to December 2012, a total of 4,125,638 questions have been asked by users on Stack Overflow, with a mean of 85,950 questions a month.

Stack Overflow provides data dumps of all user generated data, including questions asked with the list of answers, the accepted answer per question, up/down votes, favourite counts, post score, comments, and anonymized user reputation. Stack Overflow allows users to tag discussions and has a reputation-based[3] mechanism to rank users based on their active participation and contributions.

For this study, we downloaded a data dump containing data from June 2008 to March 2013. Note that Stack Overflow originated only in June 2008. Therefore, our dump includes all the questions and answers on Stack Overflow until March 2013.

2.3 Goal & Motivation

Our overall goal in this study is to *understand the common challenges and/or misconceptions among web developers*. To pursue our goal, we conducted a mixed-methods analysis on the data obtained in the Stack Overflow data dump.

While there have been other studies that have focused on assessing and improving the quality of web applications after the applications have been released [7], [29], [14], [15], there has been no systematic attempt to understand the sources of confusion and misconceptions among developers while they are building web applications. Understanding these issues is a necessary first step towards improving web application quality.

3. METHODOLOGY

In order to understand common challenges and misconceptions among web developers, we study questions as well as their accepted answers similar to prior work [4] [2], followed by a fine-grained analysis of only the accepted answer of each question. Our analyzed dataset is available for download.[4]

Our research questions are formulated as follows:

RQ1: What are the categories of topics of discussion among web developers?

RQ2: What are the hot topics related to web development in terms of importance?

RQ3: Are there temporal trends present in discussions related to web development?

RQ4: How prevalent are web-related topics in discussions related to mobile web development?

RQ5: What are the main technical challenges faced by web developers?

Prior to answering the research questions, we need to understand if Stack Overflow has sufficient related data to answer these questions. To this end, we study the total number of Stack Overflow questions that are related to client-side web development. We extract questions containing the following three tags, namely, JavaScript, HTML5 and CSS, and store them separately in three datasets. Questions containing more than one of the above mentioned tags overlap among these datasets. The number of questions in the three datasets corresponding to each tag is shown in Table 3.

[3] `http://meta.stackoverflow.com/help/whats-reputation`
[4] `http://www.ece.ubc.ca/~kbajaj/so/data.zip`

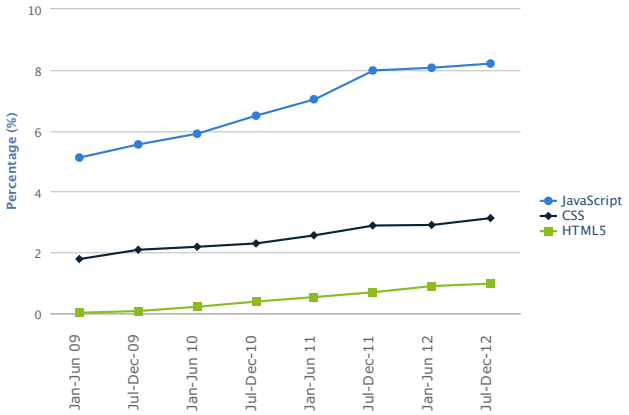

Figure 1: Share of web related questions on Stack Overflow.

Table 1: No. of questions in each subset of data.

Dataset	Tag	No. of questions	% of Questions
DS_1	JavaScript	342363	7.39%
DS_2	HTML5	31777	0.65%
DS_3	CSS	125906	2.71%
Total		**500046**	

Figure 1 shows the growth in the percentage of questions that pertain to web development over time. The results show that Stack Overflow contains a significant number of questions related to web development, from its inception. We find that the overall share of web related discussions is increasing among developers from Jan'09 to Dec'12. This indicates that *web development is gaining popularity among developers*. Further, while JavaScript continues to be the dominant topic of discussion for client-side web development (at 8%), CSS and HTML5 are gaining popularity, although their share of questions is low with 2% and 1% respectively. Therefore, these topics may gain a larger share of the questions in the future.

3.1 Data Filtering

As we saw in the previous section, there are thousands of questions related to client-side web development on Stack Overflow. In order to extract the most important questions and their answers, we devise two heuristics as follows:

- **H1:** Only accepted answers should be considered.
- **H2:** More weight should be given to questions with high view counts.

H1 is based on the analysis that majority of the accepted answers are provided by users with high reputation, and there are only 8% of users on Stack Overflow with above average reputation *(135)*. We compared the average reputation of users asking questions *(1826)* and the average reputation of users providing accepted answers *(29625)*. We found that the latter is 16 times higher than the former. From this, we conclude that *in majority of the cases, questions are asked by novice users and are predominantly answered by expert users*. Further, answers can be accepted only the users asking the question, showing that accepted answers are satisfactory from the questioner's point of view. Therefore, we consider only accepted answers to uncover important topics of discussion.

H2 is based on the fact that view count is the only statistic that is updated when a guest user views the question. Many developers use Stack Overflow to read already resolved questions. Such guest users do not actively participate in QA activities, and hence cannot affect any other statistics. Therefore, we believe questions with higher view counts are likely to be of greater interest for developers, and should be given more weight in terms of importance.

3.2 Data Processing

After we filter the dataset, we process it using Natural Language Processing (NLP) methods to understand the main topics. Alternatively, we could have used the tags to analyze the dataset. However, there are three problems with using tags for grouping: 1) tags provide only abstract information about the topic of discussion, while we want specific information, 2) the user who created a question could be unsure about the appropriate topic of discussion, thus might tag it incorrectly, 3) users tend to add as many tags as possible (up to 5) making their question visible in more search queries[5], therefore increasing the likelihood of receiving an answer quickly. Therefore, we do not use tags for the analysis.

We use Latent Dirichlet Allocation (LDA), a type of Topic Modelling to answer our research questions. Topic Modelling is a type of statistical modelling that can be used to discover hidden topics in a collection of documents, based on the statistics of words in each document [5]. LDA is a generative form of Topic Modelling that allows a set of observations to be explained by unobserved groups, that explain why some parts of data are similar [6]. The output of LDA is a list of *topics*, *topic proportion* of each document, and *topic share* of each topic in the collection. The *Topic Proportion* of each document refers to what percentage of it belongs to each topic, while the *Topic share* is a measure of how much a topic has been discussed as compared to other topics in the collection.

Figure 2 represents our overall methodology used for analyzing the questions. The rest of this section is organized according to the research questions discussed above. The steps in bold correspond to steps in Figure 2. **Step 1** and **Step 2** are common to all the analysis we do and are described in subsection 3.1. The other steps are specific to the research questions.

RQ1: Categorization of topics of discussion. To answer RQ1, i.e., listing the categories of discussions, we used LDA to categorize the discussions on Stack Overflow. Categories discovered in this phase represent major topics of discussion related to web development. We first extracted the text of questions and accepted answers **(Step 10)**. We then used the Porter Stemming Algorithm [18] to convert all words to their root words (such as "programmer" to "program") and removed stop words **(Step 11)**. Finally, we passed the resulting text as an input to LDA process **(Step 12)** for discovering hidden topics. We used the list of *generated topics* to identify the categories of discussion, and *topic share* to obtain the proportion of the discussions belonging to each category. The labels were assigned manually by the first author of the paper based on the keywords suggested by the LDA Algorithm. We have made the labels publicly available along with the dataset.

RQ2: Finding hot topics of discussion. To answer RQ2, we used LDA to analyze the top 2000 most viewed questions from each category identified in RQ1. The analysis in this phase is based on the two heuristics defined in subsection 3.1. We then ranked questions based on view count **(Step 3)** and shortlisted the first 2000 questions **(Step 4)**. Then we extracted the accepted answer text for each question and processed the text using LDA to generate a list of hot topics of discussion **(Steps 10–12)**.

[5] `http://meta.stackoverflow.com/questions/164348/`

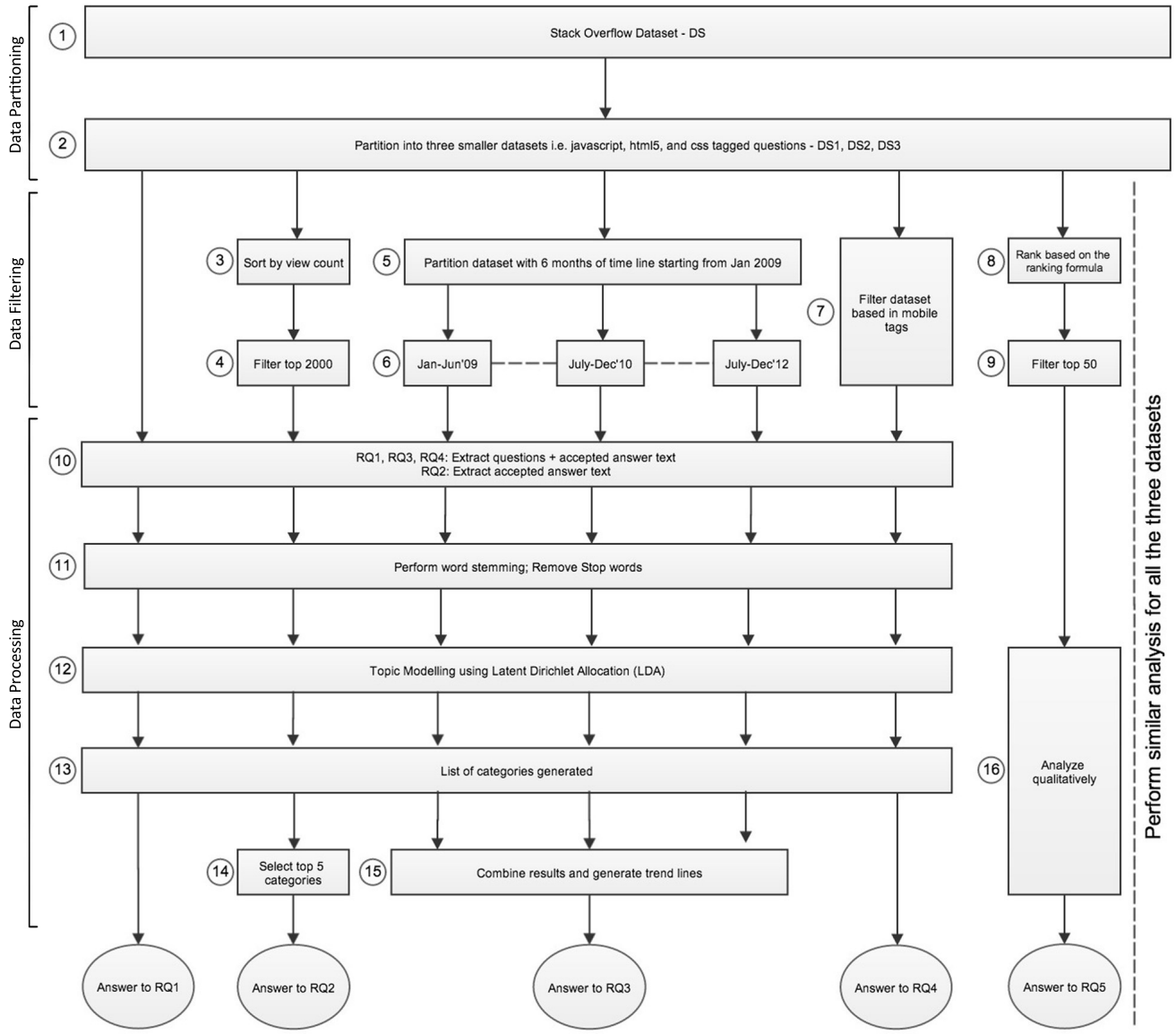

Figure 2: Our overall analysis workflow.

RQ3: Analyzing temporal trends over time. To answer RQ3, we used LDA to analyze the Stack Overflow data on a half yearly basis. We divided our dataset into subsets of 6 months data each **(Step 5)**, followed by LDA **(Steps 10–12)** to analyze important topics of discussion in each time period. The choice of 6 months was based on the trade-off between the number of questions required for efficient topic modelling versus analysis granularity. Decreasing the time period further will decrease the input data, affecting the efficiency of topic modelling. Our data spanned from July 2008 to March 2013, so we considered 8 subsets each for DS_1, DS_2, and DS_3 starting from Jan'09-Jun'09 till July'12-Dec'12. We decided to skip the first 6 months of the data as the number of questions on Stack Overflow were limited during that time period, since the site had just been launched.

RQ4: Prevalence of web in mobile development. To answer RQ4, we first analyzed the trend of JavaScript, CSS and HTML5 related discussions within the subset of questions related to mobile development. We then created subsets of these questions and performed LDA on those datasets. We wanted to study the categories of discussion related to mobile development and whether these categories are different from those in web development.

To filter out questions related to mobile development, we relied on mobile platform specific tags used by the users. The usage of tags for filtering the questions is justified as we are using the generic tags (as described in Section 3.2) that differentiate between different mobile development platforms. We used the mobile development related tags shortlisted in prior work [10] to a create subset of the data **(Step 7)** from the three datasets that we had. The tags are: *android*, *bada*, *blackberry*, *iphone*, *ios*, *java-me*, *phonegap*, *symbian*, *tizen*, *webos*, and *windows-phone*. Next we performed LDA **(Step 10–12)** to identify the main topics of discussion.

RQ5: Technical challenges faced by developers. To answer RQ5, we first select important questions and qualitatively analyze them in depth. To select the important questions, we devise a metric based

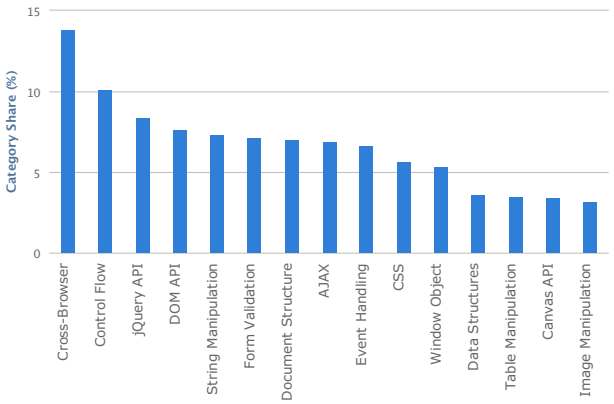

Figure 3: Categories of JavaScript-based discussions.

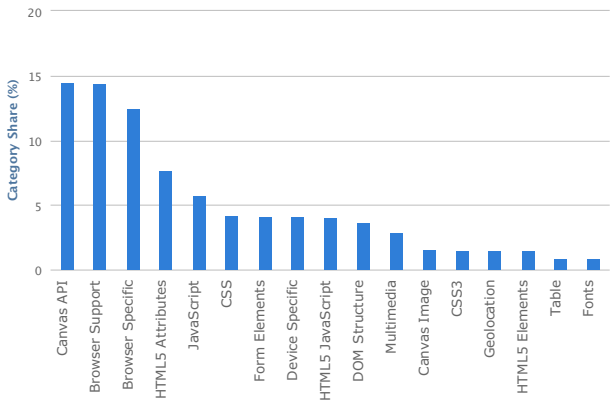

Figure 4: Categories of HTML5-based discussions.

on the statistics provided by Stack Overflow, and rank the questions **(Step 8)**. The reason we need a new metric is that the metrics used by Stack Overflow do not necessarily indicate the question's importance. For example, Stack Overflow provides a post score which is the sum of the up votes for a post minus sum of the down votes. However, the votes accrued by a question do not differentiate the number of users involved in the discussion from those who are just interested in the solution. This is important as users who are involved in a discussion may have a very different perspective from users who simply view the solution, and up/down vote the answer. Further, the reputation of the user who votes on a question is also important.

To estimate a question's importance taking the above factors into account, we propose a new metric, called *Accumulated Post Score* (AMS):

$$AMS_i = 3U_i - 25D_i + 10C_i + A_i + F_i, \quad (1)$$

where U, D, C, A, and F are as presented in Table 3.2. The weights assigned to these factors are based on the value of reputation required to perform each of these activities on Stack Overflow[6].

After computing the *accumulated post score*, we filter the top 50 **(Step 9)** questions with the highest score from each dataset, and analyze them manually **(Step 16)**. We choose 50 to balance the depth of the qualitative analysis with the time taken for the analysis.

4. RESULTS

In this section, we present the results of our study, according to the research questions formulated in the previous section.

4.1 Discussion Categories

To answer RQ1, we used Latent Dirichlet Allocation on the three obtained datasets. Figures 3–5, present the results of this process corresponding to JavaScript, HTML5 and CSS3, respectively. We provide some examples related to these categories in subsection 4.5. The results in this phase provide us with an aggregate picture of the topics that have gained most attention from web developers over the past 4 years.

Figure 3 shows the distribution of topics related to JavaScript. As can be seen in the figure, *Cross Browser Compatibility* related discussions have the maximum weight among all topics. This implies that developers have faced challenges in making their code work consistently on all browsers. Further, DOM related discussions have gained significant attention from developers. This confirms the results of our previous study [15], where we analyzed bug reports from different web applications and JavaScript libraries, and found that DOM related errors were dominant. Other popular issues being discussed are event handling, form validation and the jQuery library.

We then compared our results with the JavaScript reference provided by w3schools[7] to find what topics were missing in the discovered categories. These included features such as `eval`, cookies, and navigator. This shows that developers do not have many questions or concerns about these topics. This is somewhat counterintuitive as the first two of these topics have dependability and security implications.

Figure 4 shows the distribution of topics related to HTML5. Here, the Canvas API has been a major topic of discussion among HTML5 developers. Examples of questions regarding Canvas include (1) handling images in canvas, and (2) converting HTML to Canvas and vice versa. HTML5 browser support has also been a major issue discussed among developers, such as a feature being supported by one browser but (yet) not others. Usage of new attributes such as "data-" , media elements such as audio and video, new form elements such as email input and HTML5 based form validation are other topics of discussion.

When comparing the topics discussed with the w3schools reference for HTML5, we found that there was little to no discussion related to HTML features such as Drag & Drop and Web-Workers[8].

Figure 5 shows the distribution of topics related to CSS. Among CSS topics, the layout of the DOM tree has gained the maximum attention from developers. Other common topics for discussion are (1) questions on placing an HTML element inside/outside another, (2) creating a web page that is displayed uniformly across browsers, (3) questions related to CSS Box-Model, which describes the content of the space taken by an element,(4) modifying CSS using external widgets and JavaScript code, and (5) having custom fonts on a webpage. Again, based on the w3schools reference for CSS,

[6] `http://stackoverflow.com/help/privileges?tab=all`

[7] `http://www.w3schools.com/js/default.asp`

[8] A web worker is a JavaScript module that runs in the background, independently of other scripts, without affecting the performance of the page.

Table 2: Factors used in our Accumulated Post Score formula.

Name	Definition	Rationale	Required Reputation
Answer count - A_i	Represents the number of answers provided to the question.	High number of answers implies more people are trying to figure out the correct solution to the problem.	0
Comment count - C_i	Represents the number of comments on a particular question.	High number of comments implies more people are interested in the particular topic.	50
Favourite count - F_i	Represents the number of users marked the question as favourite.	High favourite count implies more people are interested in the solution.	0
Up votes - U_i	Represents the number of people who promoted the question.	More number of people liking the question implies the topic of discussion is important to the community.	15
Down votes - D_i	Represents the number of people who demoted the question.	More number of people not liking the question implies the question is incorrect or does not provide any value to the community.	125

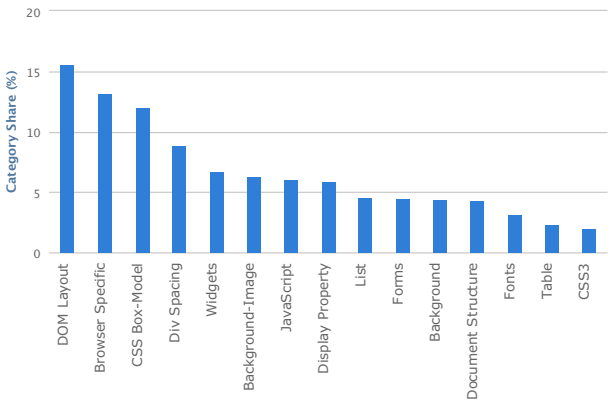

Figure 5: Categories of CSS-based discussions.

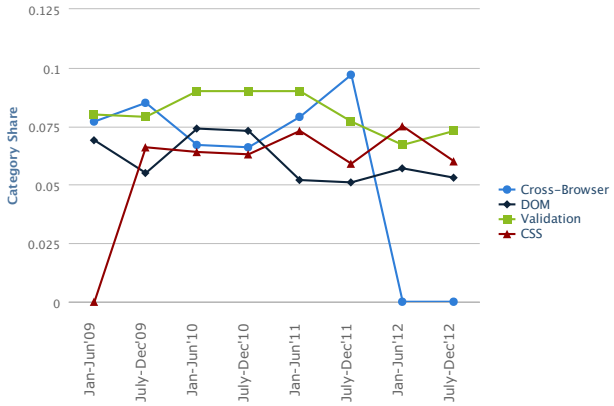

Figure 6: Temporal trends in JavaScript-based discussions.

we found that there was limited discussion on CSS features such as Sprites, i.e., a collection of images in a single image.

> ***Finding #1***: *Cross Browser related discussions have gained maximum attention from web developers, followed by DOM and Canvas related discussions.*

4.2 Hot Topics

To answer RQ2, we identify topics that have been viewed by the most number of developers regardless of how much these topics are discussed among developers (this was considered in RQ1). We call these *hot topics*, which are discussions that many developers view, probably to understand and resolve their issues, but for which there is no further discussion. Because not all users are logged into Stack Overflow when they view a discussion, it is difficult to tell which user viewed what content. Therefore, we consider only the aggregate view count rather than view-counts for different users when classifying a topic as hot.

Table 3: Hot topics with the highest view counts. Hot topics with little discussion are presented in boldface.

Technology	Hot topics
JavaScript	**Document Structure**, **File Handling**, Cross-Browser, jQuery, DOM
HTML5	**Media**, Browser Support, **HTML5 Elements**, Canvas API, **Offline Web**
CSS	**CSS3**, **Fonts**, JavaScript, Box-Model, Layout

Table 3 shows the hot topics we obtained. We expected the results to be similar to the topics that are discussed most often in RQ1. However, not all the hot topics are similar as can be seen from the table. For example, file handling in JavaScript and media in HTML5 were not among the discussion categories obtained in subsection 4.1. We believe that this is because the solutions posted are satisfactory, thus obviating the need for further discussions.

> ***Finding #2***: *View counts provide a hint towards recurrent issues faced by web developers such as those pertaining to HTML5 Elements, DOM structure, offline web, and CSS3.*

4.3 Temporal Trends

To address RQ3, we divided the four year time period of the data into six month intervals, and used topic modelling to analyze the dominant topics. Figures 6–8 present the temporal trends in the discussions for JavaScript, HTML5 and CSS, respectively.

As can be seen in Figure 6, DOM related issues have been consistently discussed over the span of 4 years. However, cross browser compatibility related discussions while dominant initially, have seen a sharp decline recently. This means issues related to browser compatibility have been reducing in importance over time. Possible explanations could be improvements in the quality of JavaScript IDE's, better JavaScript libraries that handle cross-browser issues (such as jQuery), and/or more robust browsers that follow W3C specifications. On the other hand, CSS related discussions have gained in importance in the recent years. Form validation issues have also been discussed consistently over the span of 4 years.

Figure 7 shows the temporal trends in HTML5 related discussions. Browser support has been discussed heavily among HTML5 developers. However, these have dropped in importance recently

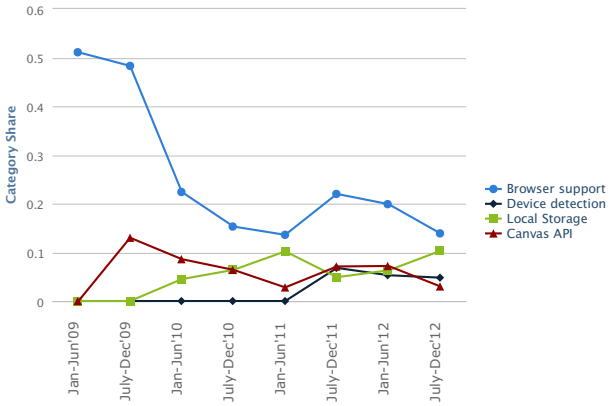

Figure 7: Temporal trends in HTML5-based discussions.

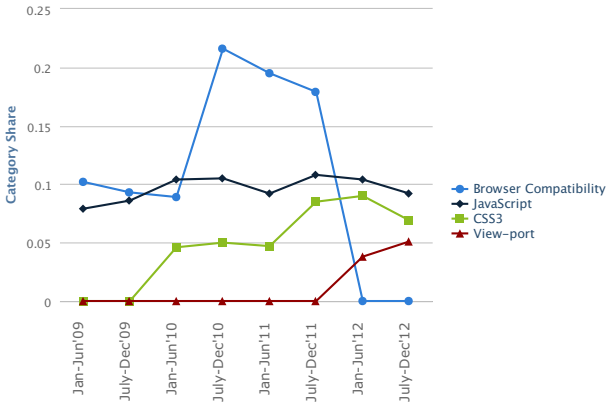

Figure 8: Temporal trends in CSS-based discussions.

suggesting that the browser support for HTML5 is maturing rapidly. The same is true for the Canvas API, which is declining in popularity. However, HTML5 specific API's such as *local storage* have gained importance over time, meaning that more and more developers are utilizing client-side storage capabilities provided by HTML5. Finally, mobile device specific issues such as interfacing device API with web applications or mobile specific themes have also become popular, suggesting that HTML5 is gaining popularity in mobile application development. The next subsection provides more details about web technologies in mobile development.

Figure 8 shows the temporal trends in CSS discussions. Again here, we can clearly observe that browser compatibility discussions have dropped sharply in the recent past. Further, JavaScript related discussions have been discussed consistently over the span of 4 years by the CSS developers, while CSS3 related discussions have increased over time. Finally, discussions related to adjusting the style of website according to the view (i.e., viewport meta tag) have recently become important, again highlighting the importance of mobile web development.

> ***Finding #3****: Cross-browser compatibility issues have seen a sharp decline in the recent past. Further, CSS3 and HTML5 discussions are gaining popularity in web as well as mobile application development.*

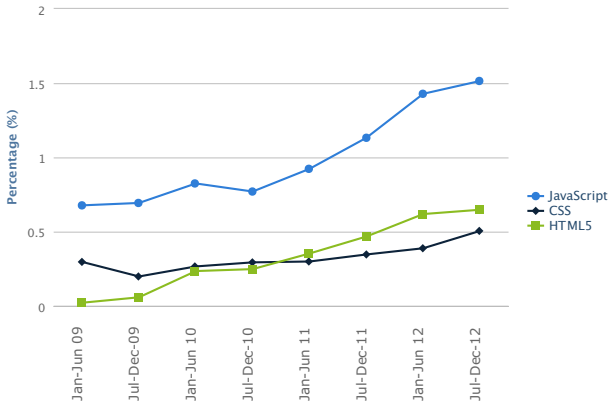

Figure 9: Share of web based discussions in mobile related questions on Stack Overflow.

4.4 Mobile Development

To answer RQ4, i.e., prevalence of web technologies in mobile development, we first study what percentage of mobile related discussions overlap with HTML5, CSS and JavaScript, over different six month time periods. As can be seen from Figure 9, the share of web based discussions is increasing within mobile related questions, although the absolute percentages are low relative to the overall share of mobile-related discussions. Further, JavaScript related discussions have seen the sharpest rise in the area of mobile development, and have nearly doubled from 0.75% to 1.5% over three years. HTML5 related discussions have gone from 0 to 0.6% in this time frame, while CSS discussions have gone from a little over 0.25% to 0.5%.

We then study the dominant topics related to JavaScript, HTML5 and CSS for mobile. We expected the results to be different from those obtained earlier, and involve mobile specific features. However, the results show that the issues are broadly similar to those in general web application development, although with some minor differences, such as geo-location and device resolution figuring prominently. Due to space constraints, we do not present the detailed results of this part of the study.

> ***Finding #4****: Discussions related to Mobile development are seeing an increasing share of web technologies such as HTML5, and follow a similar trend as in web applications.*

4.5 Technical Challenges

To gain insights into the kind of technical difficulties faced by web programmers, we ranked the questions in the three datasets based on their relative importance using Equation 1. We then manually analyzed the top 50 questions from each of the three categories, and based on the topics discussed, extracted the dominant categories in the questions. In this section, we discuss a few examples from these top 50 questions that are representative of the types of technical challenge that web developers face in their daily development activities.

Issue 1: In HTML5, developers face challenges while working with the new **HTML5-JavaScript** objects such as `localStorage`**.** For example, the following question was asked by a user on Stack Overflow:

"I'd like to store a JavaScript object in HTML5 localStorage, but my object is apparently being converted to a string. I can store and retrieve primitive JavaScript types and arrays using localStorage, but objects don't seem to work. Should they?" [9] (s.i.c)

At first glance, it seems the question is related to the datatypes that `localStorage` can store. However, the accepted answer below provides a solution combining the existing techniques used by JavaScript programmers to convert the objects into strings, showing that this was the main point of confusion for the user.

```
var testObject = { 'one': 1, 'two': 2, 'three': 3 };
localStorage.setItem('testObject',JSON.stringify(testObject));
var retrievedObject = localStorage.getItem('testObject');
console.log('retrievedObject: ', JSON.parse(retrievedObject));
```

Issue 2: Issues related to **Canvas API** are confusing for many developers. These issues vary from simple API calls to complex scripts. The following question was asked by a user with a reputation score of 13,151 on Stack Overflow, which is significantly higher than the average user reputation *(135)* on Stack Overflow, pointing to the fact that the user is an expert developer:

"Is it possible to capture or print what's displayed in an HTML canvas as an image or PDF? I'd like to generate an image via canvas, and I'd like to be able to generate a PNG from that image." [10] (sic)

The accepted answer (below) provided for this question is a simple call to one of the canvas API functions.

```
var canvas = document.getElementById("mycanvas");
var img    = canvas.toDataURL("image/png");
document.write('<img src="'+img+'"/>');
```

Questions such as this clearly indicate that there is a lack of proper and clear API documentation for HTML5. We manually analyzed the HTML5 documentation provided by W3C and inferred that it is void of many details that developers would need on a daily basis.

Issue 3: HTML5 Developers also face **browser support** issues to make their site compatible, as the example below shows.

"I have just installed IE9 beta and on a specific site I created (HTML5) IE9 jumps to compatibility mode unless I manually tell it not to. I have tried removing several parts of the website but no change. Including removing all CSS *includes. On some other website of me it goes just fine."* [11] (sic)

A simple solution (marked as accepted answer) is to tell the browser that the site is *X-UA-Compatible* – the X-UA Compatible meta tag allows web authors to choose what version of Internet Explorer the page should be rendered as – by adding an additional meta-tag.

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge"/>
```

The above question was asked by the user with a reputation score of 24,453, which implies the user is an expert. This points to the fact that many solutions to make HTML5 sites compatible are available but not known to developers.

Issue 4: In CSS tagged discussions, a developer asked the following question:

*"I have noticed I am getting a "*CSS *Explosion". It is becoming difficult for me to decide how to best organize and abstract data within the* CSS *file."* [12] (sic)

The accepted answer provided for this question lists the set of rules that a developer should follow while creating stylesheets. We inferred from the discussion that the developer was aware of the rules, but did not follow them in fear of breaking the layout or possible performance overheads. This CSS maintenance issue has been empirically highlighted before by researchers [12], calling for better tool support.

Finding #5*: Even expert programmers get confused about features of JavaScript, HTML5, and* CSS*, suggesting that the available API resources for these features is far from ideal. Also, maintaining web code, such as CSS, is complex without proper tool support, and users often ignore recommendations and best practices.*

5. DISCUSSION

In this section, we discuss the implications of our findings for web developers, researchers, and the web standardization community. We also consider the threats to validity of our results.

5.1 Implications

Web developers can use the results to focus and learn from common issues that are discussed among other developers. Educating the developers with the common sources of misconception will avoid future errors and eventually save development time. Findings 1 and 3 suggest that while cross-browser issues were important in the past, and have been discussed extensively, they seem to be much less important nowadays. Therefore, developers can shift their focus to other issues, as solutions related to browser issues are available online. Results of subsection 4.3 suggest that we need better IDEs that can assist the developer when coding against DOM and Canvas APIs. Finding 5 suggests that we need better API resources for new features in HTML5 and JavaScript and better code maintenance tool support for CSS.

The *Research community* can use our results to focus on specific areas of web development. For example, there have been many papers on cross-browser compatibility testing [13, 21], yet it appears

[9] http://stackoverflow.com/questions/2010892
[10] http://stackoverflow.com/questions/923885
[11] http://stackoverflow.com/questions/3726357
[12] http://stackoverflow.com/questions/2253110

that this is no longer the dominant problem faced by web developers (Finding 3). Rather, the issues confronting web developers today seem to be around DOM and canvas interactions. Analyzing what features of HTML5 are gaining popularity and what features are inconvenient for developers to implement can improve the overall quality of web development. Finding 4 suggests that mobile development follows a similar trend as web applications. Therefore predicting what features of HTML5 and JavaScript will be popular in mobile applications can guide the developers to build better mobile development tools.

Application of our mining methodology is not just restricted to web related discussions. Researchers can use our methodology for analyzing any area of interest by selecting appropriate tags to create subset of data. The methodology for addressing RQ2 takes view counts into consideration. As we have seen in Finding 2, view counts provide a different perspective on the relative importance of discussion items. Further, our formula is based on statistics provided by Stack Overflow, however, it is not restricted to Stack Overflow questions. It can be used in any QA website as long as the web site provides similar statistics, with suitable modification of the weights. However, we restrict ourselves to use the factors provided by Stack Overflow and use similar weights as used by Stack Overflow.

The *web standardization community* can use the results (Finding 5) to extract the areas of web development that need improvements and prioritize them in terms of standardization. The results can be used to analyze what features are lacking in web applications, and what areas need better (API) documentation to enhance developer comprehension. The results can also be used to analyze how long it takes for a particular feature to become popular after being specified in the standard. For example, understanding what features are quickly adopted by developers, can aid the development of new features and their standardization.

5.2 Threats to Validity

An *external threat* to validity of our results is that we focus on a single website, Stack Overflow. However, Stack Overflow is one of the most popular and largest question and answer websites for software developers currently. At the same time, Stack Overflow is relatively new, having started only in 2008, and hence is not representative of all issues web developers have faced in their development endeavours.

An *internal threat* to validity is that we focus only on discussions tagged JavaScript, CSS and HTML5. However, as we have seen in Table 3, this constitutes a significant number of questions numbering in the tens of thousands. Therefore, we believe that these questions are representative of client-side web development.

A *construct threat* to validity is that we designed a new metric to rank questions by their importance (Equation 1) for qualitatively analyzing questions posed by developers (RQ5), and the fact that we did this part of the analysis manually. However, our metric is based upon statistics collected by Stack Overflow, and uses some of the relative weightings that Stack Overflow itself uses for ranking questions. The majority of our analysis was done using automated methods, and was hence unbiased.

Another *construct threat* to validity is that we base one of our heuristics (H1) on user reputation. A user with high reputation score could contribute to a certain subset of posts that he knows a lot about, but ask questions about areas and languages in which he or she would be considered a novice. However, we observed majority of the questions were asked by users with low reputation score, which means they are not exert in any area.

6. RELATED WORK

Stack Overflow has been extensively studied and analyzed for a wide variety of empirical studies. For example, researchers have used Stack Overflow to understand user behaviour [16, 20, 27, 3, 11], analyze prominent topics of discussion [30, 4, 2], extract documentation [17], assign tags to discussions [22], and analyze code [26]. Other work uses Stack Overflow data to analyze mobile API usage [9] and security issues related to these APIs [25]. However, none of these studies have analyzed web development data on Stack Overflow, particularly for the client-side. As we have seen in this paper, such discussions are increasing in volume and hence it is important to understand them. To the best of our knowledge, we are the first to mine and analyze web development related discussions on Stack Overflow.

Several studies have empirically analyzed the reliability, security and performance of client-side web applications. For example, Ocariza et al. [14] used error messages logged to the console to analyze JavaScript errors in web applications. In a recent empirical study [15], we analyzed bug reports of twelve open source applications to understand the root causes of failures in them. Ratanaworabhan et al. [19] study the dynamic behaviour and performance of JavaScript-based web applications. Other work has studied the prevalence of security vulnerabilities in JavaScript such as Cross Site Scripting [28]. The main difference between these studies and ours is that we study the sources of difficulty, confusion, and misconception in programmers' minds during web application development activities. Because we analyze the natural language text of programmers questions and accepted answers, we can get to the root of a confusion or difficulty, which is typically not apparent from the code or other artifacts produced during the development process.

7. CONCLUSIONS

In this paper, we performed an empirical analysis of web related discussions on Stack Overflow, a popular question and answer forum, to understand the common difficulties and misconceptions among developers. Our study involves analyzing the text of both questions and answers related to web development to extract the dominant topics of discussion using topic modeling.

Our results show that (1) cross-browser related discussions while prevalent in the past, are becoming less important, (2) DOM APIs and event handling issues have been a significant source of confusion for web development, (3) HTML5 is gaining popularity in (mobile) web applications, (4) web related topics are becoming more prevalent in mobile development, though the topics are broadly similar to those in other web applications, and (5) even expert programmers are confused by some of the new features added to HTML5 and JavaScript. The results of our study can help the development and research communities to focus on the misconceptions or sources of confusion among web developers. It can also help the web standardization community understand the adoption of various standards and the factors impeding their adoption, if any.

8. ACKNOWLEDGEMENTS

This work was supported in part by a Strategic Project Grant from the Natural Science and Engineering Research Council of Canada (NSERC), a research gift from Intel Corporation, and a MITACS fellowship. We thank the reviewers of MSR'14 for their suggestions to improve the paper.

9. REFERENCES

[1] HTML5 home | Intel developer zone. http://software.intel.com/en-us/html5/home. Accessed: 2014-02-03.

[2] M. Allamanis and C. Sutton. Why, when, and what: analyzing stack overflow questions by topic, type, and code. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 53–56. IEEE Press, 2013.

[3] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide Web*, pages 95–106. International World Wide Web Conferences Steering Committee, 2013.

[4] A. Barua, S. W. Thomas, and A. E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, pages 1–36, 2012.

[5] D. M. Blei and J. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] A. Guha, S. Krishnamurthi, and T. Jim. Using static analysis for ajax intrusion detection. In *Proceedings of the 18th international conference on World wide web*, pages 561–570. ACM, 2009.

[8] I. Hickson and D. Hyatt. HTML5: A vocabulary and associated apis for html and xhtml. *W3C Working Draft edition*, 2011.

[9] D. Kavaler, D. Posnett, C. Gibler, H. Chen, P. Devanbu, and V. Filkov. Using and asking: Apis used in the android market and asked about in stackoverflow. In *Social Informatics*, pages 405–418. Springer, 2013.

[10] M. Linares-Vásquez, B. Dit, and D. Poshyvanyk. An exploratory analysis of mobile development issues using stack overflow. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 93–96. IEEE Press, 2013.

[11] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2857–2866. ACM, 2011.

[12] A. Mesbah and S. Mirshokraie. Automated analysis of CSS rules to support style maintenance. In *International Conference on Software Engineering (ICSE)*, pages 408–418. IEEE, 2012.

[13] A. Mesbah and M. R. Prasad. Automated cross-browser compatibility testing. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 561–570. ACM, 2011.

[14] F. Ocariza, K. Pattabiraman, and B. Zorn. JavaScript errors in the wild: An empirical study. In *Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on*, pages 100–109. IEEE, 2011.

[15] F. S. Ocariza, K. Bajaj, K. Pattabiraman, and A. Mesbah. An empirical study of client-side JavaScript bugs. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 55–64. IEEE, 2013.

[16] A. Pal, S. Chang, and J. A. Konstan. Evolution of experts in question answering communities. In *ICWSM*, 2012.

[17] C. Parnin, C. Treude, L. Grammel, and M.-A. Storey. Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. *Georgia Institute of Technology, Tech. Rep*, 2012.

[18] M. Porter. {The Porter Stemming Algorithm}. 2009.

[19] P. Ratanaworabhan, B. Livshits, and B. G. Zorn. JSMeter: Comparing the behavior of JavaScript benchmarks with real web applications. In *Proceedings of the 2010 USENIX conference on Web application development*, pages 3–3. USENIX Association, 2010.

[20] F. Riahi. Finding expert users in community question answering services using topic models. 2012.

[21] S. Roy Choudhary, M. R. Prasad, and A. Orso. X-PERT: Accurate identification of cross-browser issues in web applications. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 702–711. IEEE Press, 2013.

[22] A. K. Saha, R. K. Saha, and K. A. Schneider. A discriminative model approach for suggesting tags automatically for stack overflow questions. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 73–76. IEEE Press, 2013.

[23] D. Sin, E. Lawson, and K. Kannoorpatti. Mobile web apps-the non-programmer's alternative to native applications. In *Human System Interactions (HSI), 2012 5th International Conference on*, pages 8–15. IEEE, 2012.

[24] V. S. Sinha, S. Mani, and M. Gupta. Exploring activeness of users in QA forums. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 77–80. IEEE Press, 2013.

[25] R. Stevens, J. Ganz, V. Filkov, P. Devanbu, and H. Chen. Asking for (and about) permissions used by android apps. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 31–40. IEEE Press, 2013.

[26] S. Subramanian and R. Holmes. Making sense of online code snippets. In *Proceedings of the Tenth International Workshop on Mining Software Repositories*, pages 85–88. IEEE Press, 2013.

[27] B. Vasilescu, A. Capiluppi, and A. Serebrenik. Gender, representation and online participation: A quantitative study of stackoverflow. In *International Conference on Social Informatics. ASE*, 2012.

[28] J. Weinberger, P. Saxena, D. Akhawe, M. Finifter, R. Shin, and D. Song. An empirical analysis of xss sanitization in web application frameworks. Technical report, Technical report, UC Berkeley, 2011.

[29] Y. Zheng, T. Bao, and X. Zhang. Statically locating web application bugs caused by asynchronous calls. In *Proceedings of the 20th international conference on World wide web*, pages 805–814. ACM, 2011.

[30] Z. Zolaktaf, F. Riahi, M. Shafiei, and E. Milios. Modeling community question-answering archives. 2011.