Mining Stack Overflow for Questions Asked by Web Developers

An Empirical Study

by

Kartik Bajaj

B.Tech., VIT University, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2014

© Kartik Bajaj 2014
Abstract

Modern web applications consist of a significant amount of client-side code, written in JavaScript, HTML, and CSS. In this thesis, 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. Our results show that there is a consistent knowledge gap between the options available and options known to developers. Given the presence of knowledge gap among developers, we need better tools customized to assist developers in building web applications.
Preface

The thesis is an extension of empirical study of questions asked by web developers on Stack Overflow conducted by myself in collaboration with Professor Karthik Pattabiraman and Professor Ali Mesbah. The results of this study were published as a conference paper on June 2014 in the 11th Working Conference on Mining Software Repositories (MSR) [6]. A part of this thesis (Section 3.4) was completed as a course project for the Topics in Machine Learning course in collaboration with Professor Mehdi Moradi. I was responsible for devising the experiments, creating test cases, running the experiments, evaluating and analyzing the results, and writing the manuscript. My collaborators were responsible for guiding me with the creation of the experimental methodology and the analysis of results, as well as editing and writing portions of the manuscript.

# Table of Contents

Abstract .................................................. ii

Preface ................................................... iii

Table of Contents .......................................... iv

List of Tables ............................................... vii

List of Figures .............................................. viii

Acknowledgements .......................................... x

Dedication .................................................. xi

1 Introduction ............................................. 1
   1.1 Objectives ........................................... 3
   1.2 Thesis Contribution ................................... 4
   1.3 Thesis Organization .................................. 6

2 Background on Web Applications and Related Work .... 7
   2.1 Web Applications ...................................... 7
   2.2 Stack Overflow Dataset ............................... 9
# Table of Contents

2.3 Related Work ........................................... 11
  2.3.1 Stack Overflow Dataset .......................... 11
  2.3.2 Web Application Analysis ....................... 12

3 Experimental Methodology ................................. 14
  3.1 Research Questions ................................. 14
  3.2 Data Partitioning ................................... 15
  3.3 Data Filtering ...................................... 17
  3.4 Supervised Vs. Unsupervised Learning ............... 18
    3.4.1 Overview ...................................... 19
    3.4.2 Training Data Selection ....................... 20
    3.4.3 Building the Classifier ......................... 23
    3.4.4 Classifier Comparison .......................... 24
    3.4.5 Summary ...................................... 26
  3.5 Data Processing .................................... 27
  3.6 Summary ............................................ 33

4 Results .................................................. 34
  4.1 Discussion Categories ............................... 34
  4.2 Hot Topics .......................................... 38
  4.3 Temporal Trends .................................... 39
  4.4 Mobile Development .................................. 42
  4.5 Technical Challenges ............................... 45
  4.6 Summary ............................................ 49

5 Discussion ............................................... 50
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>No. of questions in each subset of data</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Training sets for the classifier</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Training datasets based on different features of data</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Factors used in our Accumulated Post Score formula</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Hot topics with the highest view counts. Hot topics with little discussion are presented in boldface.</td>
<td>38</td>
</tr>
<tr>
<td>A.1</td>
<td>Keywords for categories in JavaScript related questions.</td>
<td>66</td>
</tr>
<tr>
<td>A.2</td>
<td>Keywords for categories in HTML5 related questions.</td>
<td>67</td>
</tr>
<tr>
<td>A.3</td>
<td>Keywords for categories in CSS related questions.</td>
<td>68</td>
</tr>
<tr>
<td>A.4</td>
<td>Keywords for categories in mobile JavaScript related questions.</td>
<td>69</td>
</tr>
<tr>
<td>A.5</td>
<td>Keywords for categories in mobile HTML5 related questions.</td>
<td>70</td>
</tr>
<tr>
<td>A.6</td>
<td>Keywords for categories in mobile CSS related questions.</td>
<td>71</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Text shadow rendering across different browsers . . . . . . . . 2

3.1 Share of web related questions on Stack Overflow. . . . . . . 16
3.2 Number of users, questions, and accepted answers based on
the average reputation of the user . . . . . . . . . . . . . . . . . . . . . 18
3.3 Number of questions per tag . . . . . . . . . . . . . . . . . . . . . 21
3.4 Accuracy when each sample belongs to maximum of 1 class . . 24
3.5 Accuracy when each sample belongs to maximum of 2 classes 25
3.6 Accuracy when each sample belongs to maximum of 3 classes 25
3.7 Our overall analysis workflow . . . . . . . . . . . . . . . . . . . . 28

4.1 Categories of JavaScript-based discussions. . . . . . . . . . . 35
4.2 Categories of HTML5-based discussions. . . . . . . . . . . . . 36
4.3 Categories of CSS-based discussions. . . . . . . . . . . . . . . . . . 37
4.4 Temporal trends in JavaScript-based discussions. . . . . . . . 39
4.5 Temporal trends in HTML5-based discussions. . . . . . . . . . 40
4.6 Temporal trends in CSS-based discussions . . . . . . . . . . . . 41
4.7 Share of web based discussions in mobile related questions on
Stack Overflow . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.8 Categories of mobile JavaScript-based discussions . . . . . . . 44
List of Figures

4.9 Categories of mobile HTML5-based discussions. . . . . . . . 44
4.10 Categories of mobile CSS-based discussions. . . . . . . . . 45
Acknowledgements

First of all, I would like to thank my advisors Karthik Pattabiraman and Ali Mesbah for their unwavering support. The months I spent as a Master student were some of the most intellectually and professionally rewarding I have ever had, and much of it is attributable to Karthik and Ali for constantly motivating me to think more critically and get my ideas across more effectively. Their exemplary supervision allowed me to learn a lot and paved the way for me to begin my professional engineering career.

I would also like to thank my colleagues in CSRG for their help and critical feedback on my work. I particularly would like to thank my lab mates who always manage to make me laugh and made this entire experience enjoyable. None of this would have been possible if it had not been for the unconditional love accorded to me by my family and friends. I would like to thank my dad Ghansham, my mom Neelam, my siblings Vikram and Tanu, and all my relatives for encouraging me to do my best. I would also like to thank MITACS for giving me the opportunity to work with highly reputed professionals and also for funding opportunities provided by MITACS.

Last but not the least, I would like to thank God for giving me the chance to pursue this wonderful profession, and for guiding me throughout the process.
Dedication

To my friends and family
Chapter 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\(^2\) (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 \([26]\), 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 \([22]\). Finally, with the advent of HTML5 \([16]\), 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.

\(^1\)In this paper, when we say web development, we mean client-side web development, unless we say otherwise.

\(^2\)The Document Object Model is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. The nodes of every document are organized in a tree structure, called the DOM tree.
Chapter 1. Introduction

JavaScript is loosely typed and allows runtime creation and execution of code, which makes the web applications prone to errors. Browsers tend to handle these errors, however each browser has its own exception handling mechanism. Further, each browser has its own interpreter and renderer for CSS stylesheets. Even though majority browsers tend to render a similar layout, there are minor differences among these browsers. Figure 1.1 provides an example of such cross-browser issues where each browsers renders the text-shadow differently. The article [2] provides a brief overview of major cross-browsers issues that take up significant portion of developer time.

Mobile development has also been gaining attention among web developers [17]. The developers not only tend to develop applications to provide a unified experience among different browsers, but also provide a simplified and user-friendly experience on mobile devices such as iOS and android.

Our overall goal is to bridge the knowledge gap between the developer and research communities, and help in developing tools that will increase the
1.1 Objectives

In this study, our main goal in this study is to understand the common challenges and/or misconceptions among web developers. To pursue our goal, we conducted a quantitative as well as qualitative study of more than 500,000 Stack Overflow questions related to web development. Stack Overflow is a question and answer (QA) site for programmers. Being one of the active QA sites, 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

http://stackoverflow.com
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 [7], [3], [29], or used Stack Overflow statistics to analyze user behaviour [33], [27], [4]. 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.

1.2 Thesis Contribution

This thesis makes the following main contributions:

1. We define novel heuristics for analyzing Stack Overflow questions based on participating user reputation; this is needed since existing ways of ranking questions do not satisfy our criteria for extracting important information (Section 3.3);

2. We compare the quality and accuracy achieved by the supervised learning algorithm for automated tagging;

3. We categorize related discussions into multiple categories based on the dominant topics in the interactions among developers. We then high-
light the important topics of discussion from these categories. In addition, we categorize discussions by mobile web developers with respect to HTML5, CSS and JavaScript based on the dominant topics of discussion;

4. We identify temporal trends in related discussions to understand current and future trends in the areas pertaining to client side web development; and

5. 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. Overall share of mobile development is increasing and 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, CSS3 and JavaScript.
1.3 Thesis Organization

This chapter serves to establish the overarching goal and motivation of this thesis. Chapter 2 discusses the background information on web applications and establishes the motivation behind this work. Chapter 3 describes in detail the experimental methodology used to mine the questions and answers from the Stack Overflow website. Chapter 4 provides a detailed information regarding the results found from this study. Chapter 5 discusses the implications these results can have with respect to web developers, tool developers and the research community, as well as it lists the threats to validity for the experimental methodology. Chapter 6 discusses the related work, and Chapter 7 concludes and presents future research directions.
Chapter 2

Background on Web Applications and Related Work

This chapter provides background information about modern web applications, followed by a brief description about Stack Overflow and its data dumps. Finally we provide an overview of the related work and where we stand with respect to it.

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 [37]. 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. Users can ask questions, provide answers to the questions asked, mark the questions as favourite, up vote / down vote an answer, tag questions, and carry out other community related tasks. It has been actively used by programmers to ask questions [38]; 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 85K questions per 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 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.

The data dump consists of six files in XML format:

1. **Posts.xml**: Contains all questions and answers posted on Stack Overflow. Each question or answer is stored as a separate post with a unique id, and other attributes such as user id, time, text associated with it.

2. **Posthistory.xml**: Contains all edits made to each post.

3. **Users.xml**: Contains an anonymized list of Stack Overflow users.

4. **Comments.xml**: Contains the list of comments made to each post.

5. **Badges.xml**: Contains a list of badges earned by the Stack Overflow users[5]

6. **Votes.xml**: Contains the count for all the up votes, down votes, favourites, etc for each post.

In addition to the above, Stack Overflow makes the following meta-data available for analysis.

**Tags.** Stack Overflow allows users to tag each question, with up to a maximum of 5 tags. Users can select an existing tag provided in the autocomplete text box or create a new one. To create a new tag, users need to have a minimum level of reputation on Stack Overflow[6] This makes sure that new tags are only created by expert users, maintaining consistency among tags found on Stack Overflow. Expert users can also change the question tags, if the questions are incorrectly tagged.

**User Reputation.** Stack Overflow provides a metric called Reputation[7] to rank their users. Reputation is an approximate measurement of how much the community trusts a user; it is earned when the peers appreciate what a user is contributing. Users do not need reputation for basic site functionalities such as asking questions and providing answers, however users

---

[5] Besides gaining reputation with questions and answers, users receive badges for being especially helpful.
with high reputation score gain more privileges. The primary way to gain reputation is by posting good questions and useful answers. Votes on these posts cause user to gain (or sometimes lose) reputation. The maximum number reputation points that can be earned in a day is 200, thus making sure that the reputation gained by a user is by actively and consistently participating in the site activities.

2.3 Related Work

2.3.1 Stack Overflow Dataset

Dataset Analysis. Stack Overflow has been extensively studied and analyzed for a wide variety of empirical studies. For example, researchers have used Stack Overflow to analyze prominent topics of discussion [3, 7, 47], extract documentation [29], assign tags to discussions [35], and analyze code [42]. Saxe et al. [36] compared the code snippets found on Stack Overflow website with the malware code snippets. The preliminary results indicate that there exist similarity in naming conventions and further analysis can be used to build malware detection patterns.

Stack Overflow dataset has also been used to generate API documentation using crowdsourcing techniques. Campbell et al. [11] compared the PHP and Python code snippets found on Stack Overflow website with the official documentation to discover topics that are inadequately covered by the API documentation. Parnin et al. [29] investigates the dynamics of a successful API community on Stack Overflow.

Metadata Analysis. Other work uses the metadata (such as user age,
knowledge level, gender, tags for each question) attached to the Stack Overflow questions to understand user behaviour [4, 21, 27, 33, 44]. Prior work has also used Stack Overflow metadata to analyze mobile API usage [19] and security issues related to these APIs [41] by analyzing tags attached to each question.

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.

### 2.3.2 Web Application Analysis

**Empirical Studies.** Several studies have empirically analyzed the reliability, security and performance of client-side web applications. For example, Ocariza et al. [26] used error messages logged to the console to analyze JavaScript errors in web applications. In a recent empirical study [25], we analyzed bug reports of twelve open source applications to understand the root causes of failures in them. Ratanaworabhan et al. [32] study the dynamic behaviour and performance of JavaScript-based web applications. Nikiforakis et al. [24] performed an empirical study to analyze the trust relationship between the production website and JavaScript library providers.

**Bug Finding.** Prior work has also focused in finding bugs in existing web applications. Bleaker et al. [8] developed an automated technique for finding

\textbf{Security Vulnerability Analysis.} Other work has studied the prevalence of security related vulnerabilities in JavaScript code such as Cross Site Scripting [45]. Cova et al. [12] analyzed JavaScript code to detect drive-by-download attacks and malicious JavaScript code within the production web application. Dang et al. [18] performed an empirical study to analyze privacy violating information flows within existing web applications.

The \textbf{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.
Chapter 3

Experimental Methodology

In this chapter, we list the research questions in our experiments. We then provide an overview of the Stack Overflow dataset available for our analysis. Next, we describe data filtering heuristics used in our experiments. Finally, we explain what class of learning algorithms is a best fit for our analysis and how we process the available dataset, with respect to each research question.

3.1 Research Questions

Our research questions are formulated as follows:

RQ0: What percentage of the questions on Stack Overflow are related to client side web development and how is that share changing with time?

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?
3.2. Data Partitioning

Table 3.1: No. of questions in each subset of data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Tag</th>
<th>No. of questions</th>
<th>% of Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS&lt;sub&gt;1&lt;/sub&gt;</td>
<td>JavaScript</td>
<td>342363</td>
<td>7.39%</td>
</tr>
<tr>
<td>DS&lt;sub&gt;2&lt;/sub&gt;</td>
<td>HTML5</td>
<td>31777</td>
<td>0.65%</td>
</tr>
<tr>
<td>DS&lt;sub&gt;3&lt;/sub&gt;</td>
<td>CSS</td>
<td>125906</td>
<td>2.71%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>500046</td>
<td></td>
</tr>
</tbody>
</table>

**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?

### 3.2 Data Partitioning

Prior to answering the research questions, we need to understand if Stack Overflow has sufficient related data to answer these questions. Being one of the active QA sites [38], developers for all languages are attracted to Stack Overflow. 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 3.1.

Figure [3.1](#) 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
3.2. Data Partitioning

Figure 3.1: Share of web related questions on Stack Overflow.

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.

**Finding #0**: Approximately 10% of Stack Overflow questions belong to client side web development and this share is consistently increasing with time.
3.3 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 by 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 as a reference for common questions and answers, leading to high view counts. Therefore, we assign more weight to questions with high view counts.
3.4 Supervised Vs. Unsupervised Learning

Figure 3.2: Number of users, questions, and accepted answers based on the average reputation of the user

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.

3.4 Supervised Vs. Unsupervised Learning

A straw man approach to gain insights from the available dataset would be to count the tags attached to each question. However, there are three problems with using tags for grouping: 1) tags provide only abstract information about the topic of discussion, whereas 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.
3.4. Supervised Vs. Unsupervised Learning

(Up to 5) making their question visible in more search queries\(^8\) therefore increasing the likelihood of receiving an answer quickly. Therefore, we cannot use tags for the analysis and need to utilize machine learning techniques to gain insights from the dataset.

Before we further dive into the details of our methodology, we first need to understand which class of machine learning algorithms (supervised vs. unsupervised) is a good fit for our analysis. Supervised learning takes place when the training instances are labelled with the correct result, which gives feedback on how learning is progressing. This is akin to having a supervisor who can tell the agent whether it was correct. In unsupervised learning, the goal is harder because there are no pre-determined categorizations.

3.4.1 Overview

The overall goal of this phase is to see if we can learn from the tags attached to each question, and build an automated tagging system that can then be used to rectify and improve the tags attached to each question in the dataset. There has been a vast of prior work done that focus on building automated tagging systems [13, 15, 28, 39, 40]. The quality of such automated tagging can vary depending upon various aspects of training data set, such as number of classes (tags), number of samples per class (questions per tag), number of classes a sample belongs to (tags per question) and the quality of samples (length of each question). Therefore, in order to gain better insights on the quality of an automated tagging system, we select our training samples based on the criteria described in the following section.

\(^8\)http://meta.stackoverflow.com/questions/164348/
3.4. Supervised Vs. Unsupervised Learning

For each training set, we then use Naive Bayes Classifier to build an automated tagging system (as it is one of the simplest and efficient classifier for text classification [43]) followed by a 10-Fold Cross Validation Technique\(^9\) to measure the accuracy for each system. Next, we analyze the results obtained from each classifier and compare the effects of different features used to select the training data.

3.4.2 Training Data Selection

The dataset provided by Stack Overflow is huge. Directly processing all the questions and their tags will be time consuming. Therefore we select a subset of data to perform our analysis. We use the DS\(_1\) (Table 3.1) dataset defined in section 3.2 as it contains 300,000+ questions available for analysis. Since all the questions in this dataset had one common tag javascript attached to them, i.e., all questions belong to same class, any classifier we build will tag the new question to this class. In order to remove this bias we removed this tag from all the questions. We also removed the questions that had only one tag(javascript) attached to them, so that there is no question with 0 tags after removal of javascript tag i.e each sample belongs to at least one class.

As the quality of automated tagging system can vary depending upon the number of classes each sample belongs to, we created 3 initial sets of training data. Each training set consisted of tags as classes and question texts as training samples. For Training Set 1, we only considered one tag attached to each question. This way each question belongs to one and only one tag, i.e., each training sample belongs to a single class. For Training

Set 2, we considered maximum of 2 tags attached to each question, this way each training sample can belong to one or two classes. For Training Set 3, we considered three tags attached to each question. From all the three sets, we removed the tags that had less than 500 questions attached to them. This step was to make sure that we have enough training samples for each class. Distribution for the number of questions per tag in each training set can be found in Figure 3.3. The area under each curve represents the total number of questions in each dataset.

Table 3.2 provides the total number of training samples and number of classes in each dataset. To gain better insights depending upon the various factors, we created subsets of data that filter the training samples based on different requirements. One such requirement was minimum number of
3.4. Supervised Vs. Unsupervised Learning

Table 3.2: Training sets for the classifier

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Number of Tags</th>
<th>Total number of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set 1</td>
<td>41</td>
<td>161304</td>
</tr>
<tr>
<td>Training Set 2</td>
<td>82</td>
<td>258696</td>
</tr>
<tr>
<td>Training Set 3</td>
<td>114</td>
<td>306149</td>
</tr>
</tbody>
</table>

characters in each question i.e., quality of training samples. So we created 3 subsets for each of these with different minimum question length i.e. 1000, 1500, 2000 characters. Further, for each of these datasets we created 4 separate datasets, where each training set had a different requirement for minimum number of questions (500, 1000, 1500, and 2000) per tag i.e. number of samples per class. So in total we had $3 \times 4 \times 3 = 36$ different datasets for analysis.

Table 3.3: Training datasets based on different features of data

<table>
<thead>
<tr>
<th>No. of tags</th>
<th>Tag frequency</th>
<th>Question length</th>
<th>No. of shortlisted tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>500</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>1000</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>1500</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>2000</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1500</td>
<td>500</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>1500</td>
<td>1000</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1500</td>
<td>1500</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1500</td>
<td>2000</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2000</td>
<td>500</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2000</td>
<td>1000</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2000</td>
<td>1500</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2000</td>
<td>2000</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>500</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>1000</td>
<td>14</td>
</tr>
</tbody>
</table>

continued on next page...
3.4. Supervised Vs. Unsupervised Learning

Table 3.3 – continued from previous page

<table>
<thead>
<tr>
<th>No. of tags</th>
<th>Tag frequency</th>
<th>Question length</th>
<th>No. of shortlisted tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1000</td>
<td>1500</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>2000</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>500</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>1500</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>2000</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>500</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>1000</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>1500</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>2000</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>500</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1000</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1500</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>2000</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>500</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>1000</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>1500</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>2000</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>500</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>1000</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>1500</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2000</td>
<td>2000</td>
<td>3</td>
</tr>
</tbody>
</table>

3.4.3 Building the Classifier

Table 3.3 provides an overview of each training dataset. As seen in the table, the number of shortlisted tags differ. The more the number of tags, the better is the quality of automated classification. However, as noticed from the table, to include more tags we need to compromise on the quality of training data by including classes with less number of samples, and samples with lower quality, which in return will affect the accuracy of the automated classifier.
3.4. Supervised Vs. Unsupervised Learning

Therefore, we can say that there exist a balance between the accuracy and the quality of suggestions provided by the classifier.

3.4.4 Classifier Comparison

We used Naive Bayes Classification to train our automated tagging system using the different training sets defined in Table 3.3. To measure the accuracy of each classifier we used 10-Fold Cross Validation Technique. We then compared the results in terms of accuracy of each classifier with respect to quality of samples, minimum number of samples per class, and maximum number of classes per sample.

Figure 3.4 - 3.6 provide an overview of the accuracy of each classifier with respect to the parameters described above. As we can see from the results, the accuracy of each classifier varies depending upon the quality of training samples. The better the quality of training data, the higher is the accuracy.
### 3.4. Supervised Vs. Unsupervised Learning

<table>
<thead>
<tr>
<th>Quality of Samples (Questions length)</th>
<th>Accuracy of Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum 500 Characters</td>
<td>Minimum 1000 Samples</td>
</tr>
<tr>
<td>Minimum 1000 Characters</td>
<td>Minimum 1500 Samples</td>
</tr>
<tr>
<td>Minimum 1500 Characters</td>
<td>Minimum 2000 Samples</td>
</tr>
<tr>
<td>Minimum 2000 Characters</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimum 500 Characters</th>
<th>Minimum 1000 Characters</th>
<th>Minimum 1500 Characters</th>
<th>Minimum 2000 Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Samples</td>
<td>Accuracy of Classifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum 500 Characters</td>
<td>Minimum 1000 Samples</td>
<td>Minimum 1500 Samples</td>
<td>Minimum 2000 Samples</td>
</tr>
</tbody>
</table>

#### Figure 3.5: Accuracy when each sample belongs to maximum of 2 classes

#### Figure 3.6: Accuracy when each sample belongs to maximum of 3 classes
3.4. Supervised Vs. Unsupervised Learning

Based on the accuracy of each classifier, the key insights are as follows:

- Increasing the number of minimum samples per class increases the accuracy of the automated classifier, whereas the quality of the designed classifier is decreased due to lesser number of classes available for classification.

- Increasing the quality of samples also increases the accuracy of the accuracy of the automated classifier, whereas the quality of the designed classifier is reduced due to lesser number of qualified samples, leading to lesser number of qualified classes for classification.

- The accuracy of the classifier is reduced when a single sample belongs to more than one class.

3.4.5 Summary

As a result of our analysis, we conclude that the desired accuracy for an automated tagging system can be achieved. However increasing the accuracy decreases the quality of the tagging system. In our following experiments, we need to analyze the complete dataset to understand the discussions among web developers. Therefore, building a classifier using supervised learning techniques will result in low accuracy, as the quality of samples as well as the number of samples per class vary over a broad range starting from 1 sample per class. Hence, for our analysis we rely upon unsupervised learning methods, where we do not have any prior information about the classes and each sample is treated independent of the tags attached to it.
3.5 Data Processing

In order to understand common challenges and misconceptions among web developers, we study questions as well as their accepted answers similar to prior work [7] [3], followed by a fine-grained analysis of only the accepted answer of each question. Our analyzed dataset is available for download.[10]

After we filter the dataset, we process it using Natural Language Processing (NLP) methods to understand the main topics. 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 [9]. 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 [10]. 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 3.7 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 3.7. Step 1 and Step 2 are common to all the analysis we do and are described in section 3.3. The other steps are specific to the research

http://www.ece.ubc.ca/~kbajaj/so/data.zip
3.5. Data Processing

Figure 3.7: Our overall analysis workflow.
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 [31] 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 author 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 section 3.3. 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 (Heuristic 1) for each question and processed the text using LDA to generate a list of hot topics of discussion (Steps 10–12).

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 3.5) that differentiate between different mobile development platforms. We used the mobile development related tags shortlisted in prior work [20] 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 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):

\[ AMS_i = 3U_i - 25D_i + 10C_i + A_i + F_i, \]  
(3.1)

where \( U, D, C, A, \) and \( F \) are as presented in Table 3.4. The weights assigned to these factors are based on the value of reputation required to perform each of these activities on Stack Overflow.\[11\]

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.

\[ \text{http://stackoverflow.com/help/privileges?tab=all} \]
Table 3.4: Factors used in our Accumulated Post Score formula.

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

In this chapter, we presented our technique to analyze the Stack Overflow discussions. We first formulated two heuristics to rank questions based on the number of views attained by each question. We then compared the quality and accuracy achieved using supervised learning algorithms. Based on our analysis, we chose unsupervised learning to answer our research questions. Next, we defined our methodology to answer each research question. Last, we defined *Accumulated Post Score* metric to rank Stack Overflow questions based on the quality of discussion generated by each question. In the following chapter, we present the results of our analysis with respect to each research question.
Chapter 4

Results

In this chapter, we present the results of our study, according to the research questions formulated in the Section 3.1. For each research question, we describe the outcome of our methodology as well as summarize the results in terms of findings. The sections in this chapter correspond to the research questions.

4.1 Discussion Categories

To answer RQ1, we used the LDA method in the previous chapter on the three obtained datasets. Figures 4.1–4.3 present the results of this process corresponding to JavaScript, HTML5 and CSS3, respectively. We provide some examples related to these categories in section 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 four years.

Figure 4.1 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 de-
4.1. Discussion Categories

Developers. This confirms the results of our previous study [25], 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\footnote{http://www.w3schools.com/js/default.asp} 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 4.2 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 can-
Figure 4.2: Categories of HTML5-based discussions.

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.\footnote{A web worker is a JavaScript module that runs in the background, independently of other scripts, without affecting the performance of the page.}

Figure 4.3 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 plac-
4.1. Discussion Categories

![Bar Chart]

Figure 4.3: Categories of CSS-based discussions.

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

A detailed list of keywords attached to each category for web related discussion in JavaScript, HTML5 and CSS can be found in the Table A.1–A.3 in the appendix.

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

Table 4.1: Hot topics with the highest view counts. Hot topics with little discussion are presented in boldface.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Hot topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript</td>
<td>Document Structure, File Handling, Cross-Browser, jQuery, DOM</td>
</tr>
<tr>
<td>HTML5</td>
<td>Media, Browser Support, HTML5 Elements, Canvas API, Offline Web</td>
</tr>
<tr>
<td>CSS</td>
<td>CSS3, Fonts, JavaScript, Box-Model, Layout</td>
</tr>
</tbody>
</table>

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 4.1 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 section 4.1. We believe that this is because the solu-
4.3 Temporal Trends

Figure 4.4: Temporal trends in JavaScript-based discussions.

As can be seen in Figure 4.4, 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 re-

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 within each subset of data. Figures 4.4–4.6 present the temporal trends in the discussions for JavaScript, HTML5 and CSS, respectively.

As can be seen in Figure 4.4, 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 re-
4.3. Temporal Trends

Figure 4.5: Temporal trends in HTML5-based discussions.

cently. 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 4.5 shows the temporal trends in HTML5 related discussions. Browser support has been discussed heavily among HTML5 developers. However, these have dropped in importance recently 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
4.3. Temporal Trends

Figure 4.6 shows the temporal trends in CSS-based 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 impor-
tance 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.

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 4.7, 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.
4.4. Mobile Development

Figure 4.7: Share of web based discussions in mobile related questions on Stack Overflow.

Figure 4.8 shows JavaScript discussions related to mobile development. As can be seen in Figure 4.8, JavaScript discussions related to topics such as file handling and event handling are prevalent in mobile development, which was also the case for web applications. However, mobile specific discussions such as device API, device resolution, and geolocation are also prevalent, but not found in web development discussions.

Figure 4.9 shows HTML5 discussions related to mobile development. As was the case in web applications, browser support related discussions are prevalent in mobile applications, as are discussions related to media/form elements. However, mobile specific issues such as Touch events are also prevalent, which was not the case for web applications.

Figure 4.10 shows CSS discussions related to mobile development. Mobile-specific issues such as device resolution, zooming and touch sensitivity are
4.4. Mobile Development

Figure 4.8: Categories of mobile JavaScript-based discussions.

Figure 4.9: Categories of mobile HTML5-based discussions.
prevalent among the discussions, next to generic topics such as cross-browser, box-model, and layout.

A detailed list of keywords attached to each category for mobile related discussion in JavaScript, HTML5 and CSS can be found in the Table A.4-A.6 in the appendix.

**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 3.1. We then manually analyzed the top 50 ques-
4.5. Technical Challenges

tions 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?”

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.

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


46
4.5. Technical Challenges

**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.”

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

```javascript
1 var canvas = document.getElementById("mycanvas");
2 var img = canvas.toDataURL("image/png");
3 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.

[http://stackoverflow.com/questions/923885](http://stackoverflow.com/questions/923885)
4.5. Technical Challenges

“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 mine it goes just fine.” [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.

1 `<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.” [sic]

The accepted answer provided for this question lists the set of rules

---

16 [http://stackoverflow.com/questions/3726357](http://stackoverflow.com/questions/3726357)
17 [http://stackoverflow.com/questions/2253110](http://stackoverflow.com/questions/2253110)
4.6 Summary

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 [22], 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.

4.6 Summary

In this chapter, we presented our results with respect to each research question asked in the previous chapter. The results of our analysis 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.
Chapter 5

Discussion

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

5.1 Implications for Web Developers

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 today. Therefore, developers can shift their focus to other issues, as solutions related to browser issues are available online. Results of section 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.
5.2 Implications for Research Community

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 \cite{23,34}, yet it appears 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.
5.3 Implications for Web Standardization Community

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. The results can also be used to see what features have been rarely used and discussed among developers.

5.4 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.

Another external threat is the reproducibility of our results. Stack Overflow is a growing website and hence, it might add more features that can affect the quality of our results as well as the proposed metric. However, to mitigate this issue we have publicly released the subset of Stack Over-
5.4. Threats to Validity

flow dataset that we analyzed and its results. The dataset is available for download\textsuperscript{18}.

Another external threat to validity of our results is that there might be duplicate discussions pertaining to the same category of questions. This might affect the category share of topics discovered in our analysis. However, questions are actively marked as duplicates on Stack Overflow if they are likely to be repeated and the solution is already available on Stack Overflow. We removed those questions that are marked as duplicates.

Another external threat to validity of our results is that we assume the questions were asked by web developers during the development phase of their application. However, it is possible that some of the questions might have been asked by quality analysts during the testing phase of the web application.

An internal threat to validity is that we focus only on discussions tagged JavaScript, CSS and HTML5. However, as we have seen in Table 3.1, this constitutes a significant number of questions numbering in the tens of thousands of questions per month. Therefore, we believe that these questions are representative of client-side web development.

Another internal threat to validity is that we assume the user with high reputation on Stack Overflow are experts. However, it is possible that a user is expert in one area and novice in another where he/she is asking the questions. This might affect the results of our Finding 5. However, in our preliminary analysis of Stack Overflow dataset we found that the average reputation of users asking the question verses the user providing the

\textsuperscript{18}http://www.ece.ubc.ca/~kbajaj/so/data.zip
5.4. Threats to Validity

A satisfying answer was 16 times lesser, which leads to the conclusion that in majority of the cases, questions are asked by user who have low reputation score and are not experts.

Another internal threat to validity is that we use “HTML5” and not the “HTML” tag to select the subset of questions for analysis. HTML5 is a recent advancement of HTML and is supported by majority of the latest mobile devices. This might bias the results of our study towards mobile devices.

A construct threat to validity is that we designed a new metric to rank questions by their importance (Equation 3.1) 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.

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 expert in any area.
Chapter 6

Conclusion and Future Work

In this thesis, 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 modelling.

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.
6.1 Future Work

While in this study we only analyzed top 50 questions, we plan to extend our analysis and include top 1000 questions to get better insights on the obtained results.

We also plan to look more closely into issues related to mobile development. To complement this study, we can use the bug reports filed by open source mobile application users. This comparison can provide us detailed insights on which mobile development are solved after discussion on QA websites and which ones remain unsolved.

Finally, as mentioned in Chapter 1, our main goal in this study is to understand the common challenges and/or misconceptions among web developers. Hence, we plan to utilize the results of our study and build tools and techniques that can assist the web developers in overcoming the challenges, and accelerate the web development process. One such project is Dompletion [5], which is a DOM-Aware JavaScript code completion system.
Bibliography


Bibliography


Bibliography


[24] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. You are what you include: Large-scale evaluation of


the non-programmer’s alternative to native applications. In Human
System Interactions (HSI), 2012 5th International Conference on, pages

[38] Vibha Singhal Sinha, Senthil Mani, and Monika Gupta. Exploring ac-
tiveness of users in QA forums. In Proceedings of the Tenth International
Workshop on Mining Software Repositories, pages 77–80. IEEE Press,
2013.

algorithms for social recommender systems. ACM Transactions on the

[40] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-
In Proceedings of the 31st annual international ACM SIGIR conference
on Research and development in information retrieval, pages 515–522.

[41] Ryan Stevens, Jonathan Ganz, Vladimir Filkov, Premkumar Devanbu,
and Hao Chen. Asking for (and about) permissions used by android
apps. In Proceedings of the Tenth International Workshop on Mining

[42] Siddharth Subramanian and Reid Holmes. Making sense of online code
snippets. In Proceedings of the Tenth International Workshop on Mining


Appendix A

Keywords for Each Category

The tables in the following pages list the keywords discovered in each category using Latent Dirichlet Allocation. Table A.1-A.3 represent the categories for web related discussions. Table A.4-A.6 represent the categories for mobile related discussions.
### Appendix A. Keywords for Each Category

Table A.1: Keywords for categories in JavaScript related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Browser</td>
<td>javascript ar thi browser support code js work ha make thei don web wai onli implement good veri version</td>
</tr>
<tr>
<td>Function Calls</td>
<td>function object var return thi call foo variabl obj properti method log scope prototype global console valu ar alert</td>
</tr>
<tr>
<td>jQuery</td>
<td>http net jsfiddle www html org thi jquery https en api plugin demo work jqueri google code javascript github</td>
</tr>
<tr>
<td>DOM</td>
<td>div class id function li click span ul href hide element item html var show attr find link data</td>
</tr>
<tr>
<td>String Manipulation</td>
<td>string var match return number str replace charact thi length split regex result test valu function text ar replac</td>
</tr>
<tr>
<td>Validation</td>
<td>input form type id function val submit button text valu var field document label return checkbox checked false check</td>
</tr>
<tr>
<td>Document Structure</td>
<td>script html javascript text js type document jquery page head body src load content tag http file function code</td>
</tr>
<tr>
<td>AJAX</td>
<td>data function json ajax url php callback var request post success return call result error response alert id type</td>
</tr>
<tr>
<td>Event Handling</td>
<td>event function click handler element document bind false thi target button trigger dialog alert return events fire jqueri onclick</td>
</tr>
<tr>
<td>CSS</td>
<td>px width css height style div left top color background position size posit border scroll font margin animate var</td>
</tr>
<tr>
<td>Window Object</td>
<td>window function var setttimeout location open href document url page timer return thi call hash popup true setinterval timeout</td>
</tr>
<tr>
<td>Data Structures</td>
<td>model view function app ext data var backbone render templat thi dojo extend template bind return require ko creat</td>
</tr>
<tr>
<td>Table Manipulation</td>
<td>option td select tr id row var table tabl function options text data selected column cell class val br</td>
</tr>
<tr>
<td>Canvas API</td>
<td>date var math function time return canvas svg random draw canva month ctx chart year start max point floor</td>
</tr>
<tr>
<td>Image Manipulation</td>
<td>img imag src image images var png jpg function url video id load player width height gif alt http</td>
</tr>
</tbody>
</table>
### Appendix A. Keywords for Each Category

Table A.2: Keywords for categories in HTML5 related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas API</td>
<td>thi canva ar imag wai make creat canvas draw ani time element onli object move work game set javascript</td>
</tr>
<tr>
<td>Browser Support</td>
<td>html thi web ar app browser support javascript user server mobil ani applic develop flash wai work android ha</td>
</tr>
<tr>
<td>Browser Specific</td>
<td>thi work http chrome html test firefox browser code problem page wa net ve support safari doe doesn tri</td>
</tr>
<tr>
<td>HTML5 Attributes</td>
<td>html element thi ar tag attribut content css browser page valid ani document don make wai ha onli thei</td>
</tr>
<tr>
<td>JavaScript</td>
<td>function var event document click window false getelementbyid return thi id alert button element code true call addeventlistener log</td>
</tr>
<tr>
<td>CSS</td>
<td>px div width height background left css style margin top border color position text padding font box display class</td>
</tr>
<tr>
<td>Forms</td>
<td>input type form id text label button submit field email post valu br html user class val valid select</td>
</tr>
<tr>
<td>Device Specific</td>
<td>page file load url cach html manifest user thi app cache android ajax content link php request browser server</td>
</tr>
<tr>
<td>HTML5 JS</td>
<td>file data server var upload function files send localstorage json string request websocket client php object xhr thi url</td>
</tr>
<tr>
<td>DOM Structure</td>
<td>script html text head body js type javascript title src css jquery http content meta href doctype id div</td>
</tr>
<tr>
<td>Media</td>
<td>video audio mp plai src player sourc play type html ogg file http controls id tag control sound flash</td>
</tr>
<tr>
<td>Canvas Image</td>
<td>imag image ing var canva src width height png function data images drawimage document context draw onload jpg</td>
</tr>
<tr>
<td>CSS3</td>
<td>svg transform webkit scale deg color animation rotate opacity var anim path fill http style camera gl width moz</td>
</tr>
<tr>
<td>Geolocation</td>
<td>map google http var maps org api function navigator position https googl locat geoloc coords error location geolocation en</td>
</tr>
<tr>
<td>HTML5 Elements</td>
<td>div data section page id header content role class footer ui article articl mobile theme tab icon button btn</td>
</tr>
<tr>
<td>Table</td>
<td>td option tr row table select id tabl db var php result echo data column function cell tx transaction</td>
</tr>
<tr>
<td>Fonts</td>
<td>font date time node var xsl format st size pt text pos scope frame url output family fonts bold</td>
</tr>
</tbody>
</table>
## Appendix A. Keywords for Each Category

Table A.3: Keywords for categories in CSS related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOM Layout</td>
<td>width height div thi content left set posit element overflow top position column scroll px fix float make css</td>
</tr>
<tr>
<td>Browser Specific</td>
<td>css thi ar browser html page media make javascript web wai support onli site ani differ design thei screen</td>
</tr>
<tr>
<td>CSS Box-Model</td>
<td>px width left margin height top border padding background position color bottom style solid float auto div text absolute</td>
</tr>
<tr>
<td>Div Spacing</td>
<td>div class id content br style html float left container header clear wrapper body test footer thi main title</td>
</tr>
<tr>
<td>Widgets</td>
<td>http net jsfiddle ui data tab thijquery icon css slider demo widget jquery dialog button tooltip tabs work</td>
</tr>
<tr>
<td>Background-Image</td>
<td>img background imag image png images url src repeat jpg alt http height href gif bg width class logo</td>
</tr>
<tr>
<td>JavaScript</td>
<td>function var document click event javascript script return jquery query hide window getelementbyid css id show code ready addclass</td>
</tr>
<tr>
<td>Display Property</td>
<td>span text block display align inline class line center style space vertical wrap word blah white float html div</td>
</tr>
<tr>
<td>List</td>
<td>li ul menu href item class nav list id link hov hover display html home navig dropdown color block</td>
</tr>
<tr>
<td>Forms</td>
<td>input button type label form option text id select class field submit textarea checkbox search radio email user html</td>
</tr>
<tr>
<td>Background</td>
<td>background color border box px shadow radius webkit gradient moz rgba linear top css bottom white red image solid</td>
</tr>
<tr>
<td>Document Structure</td>
<td>html css text type script http head href body link style www title content xhtml stylesheet js rel org</td>
</tr>
<tr>
<td>Fonts</td>
<td>font size em text color family weight bold serif sans arial url normal decoration helvetica px line style format</td>
</tr>
<tr>
<td>Table</td>
<td>td tr table tabl class row cell width nbsp border style align column col thbody collapse data id center</td>
</tr>
<tr>
<td>CSS3</td>
<td>webkit transform transition opacity moz deg anim animation ease rotate filter ms css opac scale transit svg alpha rotat</td>
</tr>
</tbody>
</table>
## Appendix A. Keywords for Each Category

Table A.4: Keywords for categories in mobile JavaScript related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Handling</td>
<td>file phonegap html plugin cordova js http app android folder xml www creat write code https api project local</td>
</tr>
<tr>
<td>Event Handling</td>
<td>event function document touch false adddeventlistener bind fire var click events touchstart element handler preventdefault true true return listen tap</td>
</tr>
<tr>
<td>Resolution</td>
<td>window width height tab viewport screen scale orient var ui content size device meta zoom ti thi view add</td>
</tr>
<tr>
<td>media</td>
<td>video navigator user ipad iphone userAgent android audio match agent os test var version ua window http safari ios</td>
</tr>
<tr>
<td>CSS</td>
<td>css div px style li class id width href webkit text height color left button link font ul http</td>
</tr>
<tr>
<td>Device API</td>
<td>webview uiwebview nsstring url javascript request stringbyevaluatingjavascriptfromstring method return void view navigationtype objective shouldstartloadwithrequest nsurlrequest bool deleg nsurl webviewdidfinishload</td>
</tr>
<tr>
<td>Drag n Drop</td>
<td>scroll var window node element posit document range function overflow position move span select text div offset length top</td>
</tr>
<tr>
<td>Image Manipulation</td>
<td>var img image imag camera src push data string options png array length thi return prototype base images photo</td>
</tr>
<tr>
<td>Geolocation</td>
<td>map google maps http googl api location position locat www coords marker geolocation geoloc link id latitude html code</td>
</tr>
<tr>
<td>DOM</td>
<td>document var element getelementbyid innerhtml createelement function appendchild string iframe ifram id class tag body label foo hittestresult length</td>
</tr>
<tr>
<td>Mobile SQL</td>
<td>tx db function id executesql var table transaction insert results row result return item rows values data errorcb sql</td>
</tr>
<tr>
<td>Facebook API</td>
<td>login connection frame id fb url user connect facebook post password states response redirect els ip network server app</td>
</tr>
</tbody>
</table>
### Table A.5: Keywords for categories in mobile HTML5 related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser Support</td>
<td>browser android support ios thi safari work html iphon test doe ipad mobil devic don ar version webkit onli</td>
</tr>
<tr>
<td>File Handling</td>
<td>file data load cach local html server app store storag ajax manifest json browser databas download folder save request</td>
</tr>
<tr>
<td>Media</td>
<td>video audio plai html player mp tag play file src media control work element source attribut stream controls load</td>
</tr>
<tr>
<td>Event Handling</td>
<td>function var false addeventlistener window document true funct log error call check navigator console phonegap return setttimeout alert connection</td>
</tr>
<tr>
<td>Touch Events</td>
<td>event function page click bind touch live handl preventdefault prevent button thi jqueri trigger document touchstart element dom handler</td>
</tr>
<tr>
<td>Resolution</td>
<td>width imag screen css scale viewport png icon meta height content size px media device pixel image set background</td>
</tr>
<tr>
<td>Device API</td>
<td>phonegap app camera phone cordova window open google maps navigator href github plugin map url sms https messag plugins</td>
</tr>
<tr>
<td>Form Elements</td>
<td>input type form php date text keyboard number content html email submit element id post test function picker val</td>
</tr>
<tr>
<td>Canvas</td>
<td>px canvas ctx var canva height ev webkit width top draw left context transform color style border center css</td>
</tr>
</tbody>
</table>
## Appendix A. Keywords for Each Category

Table A.6: Keywords for categories in mobile CSS related questions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Browser</td>
<td>ar mobil browser support web css devic site thei ipad good android ios safari make user iphon design version</td>
</tr>
<tr>
<td>Box-Model</td>
<td>px color webkit background border top left margin width bottom height style thi padding css box body html button</td>
</tr>
<tr>
<td>Resolution</td>
<td>media width screen px device onli max pixel queri min css iphon ratio portrait orientation devic landscape orient webkit</td>
</tr>
<tr>
<td>Zooming</td>
<td>width viewport scale meta content device initial tag maximum user thi scalable head set minimum html zoom page apple</td>
</tr>
<tr>
<td>Layout</td>
<td>div height id width class style header data page position container wrapper content px btn section left overflow background</td>
</tr>
<tr>
<td>JavaScript</td>
<td>function document var window script javascript return body style navigator js jquery getelementbyid click code onclick attr els modernizr</td>
</tr>
<tr>
<td>Image</td>
<td>background imag image img png url images icon size app src repeat jpg sprite limit path center jpeg bg</td>
</tr>
<tr>
<td>Touch</td>
<td>event touch function scroll hover touchstart preventdefault click trigger tap document touchend addeventlistener events element user touches bind nav</td>
</tr>
<tr>
<td>Form</td>
<td>input span text type select class label range webkit form field remov node focus wrap user placeholder pointer subview</td>
</tr>
<tr>
<td>DOM</td>
<td>li class menu display ul tab item block pre div href index make drop list tag inline hover space</td>
</tr>
<tr>
<td>CSS3 Animations</td>
<td>transform webkit translate transition transit anim ease px opacity move act ms property css moz hardwar rotate animation translatex</td>
</tr>
<tr>
<td>Fonts</td>
<td>font size family fonts text url ttf webfont weight format bold style normal src rule fac svg six pt</td>
</tr>
</tbody>
</table>