Mining Developer Questions about Major Web Frameworks

Zakaria Mehrab, Raquib Bin Yousuf, Ibrahim Asadullah Tahmid and Rifat Shahriyar
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
dmehrab94@gmail.com, raquib107@gmail.com, iatahmid@gmail.com, rifat@cse.buet.ac.bd

Keywords: Mining, Web Framework, Stack Overflow.

Abstract: Web frameworks are the de facto way to build web-enabled applications. Stack Overflow, being one of the leading question answering sites available, has become a helpful resource in numerous software engineering research. In this paper, we present a study of common challenges and issues among developers of two major web frameworks namely Laravel and Django by mining questions asked on Stack Overflow. We extracted the issues that the developers are most concerned about. We sorted these issues by popularity and difficulty metrics and observed the contrasting nature of difficulty and popularity. We also noted an exception that installation is a popular issue over both the frameworks and simultaneously it is also difficult to resolve. Besides, we found that about 50% issues are common over both the frameworks. Our findings would help the framework developers to understand better the need of the framework users by focusing most difficult and the most popular issues.

1 INTRODUCTION

Web development is a term comprised of developing websites, web services, and web application. In recent years, web applications require aesthetic representation as well as an efficient interaction between users and websites, leading to more complexity. According to Hevner et al. (Hevner et al., 2007), currently web engineers face three intractable problems: domain/system complexity, increased development time and cost. Web developers feel the need for modularization of concerns to accommodate big and complex applications (Ginige, 1998). Web frameworks help developers in this task by bringing a standard way to develop and implement web applications. These frameworks often help to decouple the logic portion from the view portion, namely implementing the Model-View-Controller(MVC) pattern. Moreover, frameworks enable developers to reuse design and implementation by combining the procedures of necessary tasks. According to J. Carlos et al. (Fernández-Conde and González-Calero, 2002) these frameworks have a positive effect on the project with shortened development time, reduced complexity, increased productivity, extensibility, and reliability.

Since the first appearance in the late 1990s, more than 5,000 frameworks have been released (Github, 2018d). While using these frameworks for creating web applications, developers often find themselves grounded with various problems. Developers often post questions in Stack Overflow (Exchange, 2017a), part of the Stack Exchange platform, to seek help and guidance. According to Mamykina et al. (Mamykina et al., 2011) Stack Overflow is larger than any other social Q&A forum or programming forum. Therefore, a thorough analysis of its posts can help us understand the problems faced by the web framework developers.

Created in 2008 by Jeff Atwood and Joel Spolsky, Stack Overflow is the flagship site of the Stack Exchange Network. It features questions and answers on a wide range of topics regarding computer programming and related technologies. The huge amount of web framework related posts in Stack Overflow shows the importance of studies related to the problems faced by web framework developers. In Figure 1, we have presented a graph with the percentage of posts related to eleven popular frameworks. Together these eleven frameworks measure up to more than 15% of the total questions on Stack Overflow each month in 2017.

We used Stack Overflow data dump (Exchange, 2017a) to explore the issues of web frameworks that developers face.

We picked two web frameworks, Laravel, and Django for our analysis. The reason behind choosing Laravel and Django is that both are very popular
but not well studied, unlike Javascript frameworks. Whereas Laravel and Django together have a little over 45,000 stars on their GitHub repository (Github, 2018b) (Github, 2018a), one Javascript framework, vuejs, alone has bagged over 95,000 stars on its GitHub repository (Github, 2018c). There is also the fact that searching with keyword “javascript” in (DBLP, 1993) shows 784 matches whereas “django” keyword produces 34 results (24 of which belongs to author "Django Armstrong") and ”laravel” keyword produces two results.

For studying the two frameworks, we extracted contents from all the related questions. We used them to train our model in MALLET topic modeling tool (MALLET, 2017). We extracted the appropriate, relevant topics of each question by Latent Dirichlet Allocation(LDA). Once we were able to extract the individual topic of each question, we started to answer our research questions.

RQ1 What are the issues users of these two framework face?

RQ2 What are the most popular issues asked by developers of these two frameworks?

RQ3 Which issues of these two frameworks are the most difficult?

RQ4 Are the issues of these two frameworks different?

From our experiment, we found 18 issues for Laravel and 20 issues of Django; faced by the developers. We sorted these issues by different metrics and ranked them based on their popularity and difficulty. We observed the contrasting nature of difficulty and popularity. We also noted an exception that installation is a popular issue over both the frameworks and simultaneously, it is also difficult to resolve. Also, we found that about 50% issues are common over both the frameworks. Our findings can help the developers of the frameworks to better understand the need of the users. They can categorize the needs of users to focus on the most difficult and the most popular issues. They can also find similar issues in multiple frameworks those require major improvement.

The rest of this paper is organized as follows. The next section reviews the related works. The methods for data collection, operationalization and analysis are presented next. This is followed by our findings and related discussions. The paper concludes by highlighting possible future extensions of our work.

2 BACKGROUND AND RELATED WORKS

Laravel is an MVC framework for PHP web language, released in 2011. Olanrewaju et al. (Olanrewaju et al., 2015) discussed the most famous MVC based PHP frameworks, evaluated their performance and found that Laravel outperforms other MVC frameworks, dubbed Laravel as the most suitable PHP framework for future web technology. Django is an open source web framework built on Python programming language. It was released in 2005 and quickly gained popularity due to re-usability of components; rapid development leads to the development of complex web applications in shorter time.

2.1 Related Work

We looked into works which discuss the issues that the developers face while using web frameworks. We also went through several types of research for better understanding the issues and trends among developers from the empirical analysis of Stack Overflow data.

The empirical studies using Stack Overflow data is now one of the growing trends of empirical software engineering. Several papers using Stack Overflow data has been presented. Some of them use the data to present behavioral research question about the Stack Overflow users. Others use topic modeling to categorize the discussions in the posts.

Allamanis et al. (Allamanis and Sutton, 2013) applied topic modeling on Stack Overflow questions. After associating them with programming concepts, they found that certain types of questions are associated with specific programming concepts. Treude et al. (Treude et al., 2011) analyzed data from Stack Overflow to categorize the kinds of questions that are asked and found that Q&A websites are particularly effective in code reviews and conceptual questions. Mamykina et al. (Mamykina et al., 2011) took the design lessons from Stack Overflow and found that the success of Stack Overflow is because of the quickness of a question getting answered. They also found that
the high visibility and daily involvement of the design
team of Stack Overflow help increase Stack Over-
flow’s popularity. Barua et al. (Barua et al., 2014)
aimed to analyze the actual textual content of Stack
Overflow to help the software engineering commu-
nity to better understand the thoughts and needs of
developers. They found that the topics of interest to
developers range widely from jobs to version control
systems to C# syntax. They dictated that the topics
gaining the most popularity over time are web de-
velopment, mobile applications, Git and MySQL. Li
et al. (Li et al., 2013) performed an empirical study
with 24 developers to understand the needs and chal-
 lenges developers face during the development phase.
Beyer et al. (Beyer and Pinzger, 2014) manually an-
alyzed 450 Android related posts to determine com-
mon problems developers face. Rosen et al. (Rosen
and Shihab, 2016) analyzed Stack Overflow data to
determine what mobile developers ask about by us-
 ing LDA-based topic models. They also determined
what popular mobile-related issues are the most dif-
ficult, explore platform-specific issues, and investi-
gated the types of questions mobile developers ask.
Bajaj et al. (Bajaj et al., 2014) presented a study of
common challenges and misconceptions among web
developers by mining-related questions about client-
side code, written in JavaScript, HTML, and CSS
asked on Stack Overflow. They used unsupervised
learning to categorize the mined questions and de-
 fined a ranking algorithm to rank all the Stack Over-
flow questions based on their importance.
The related works for this topic revolve around the
empirical use of Stack Overflow data to determine the
issues developers face, trends of asked topics and be-
 havior by the users. These tasks require textual con-
tent analyzing which is done mostly with topic mod-
ing studies. Though several empirical studies have
been done with the Stack Overflow data, none have
tried to determine the specific issues faced by web
framework users. We aim to fill that void by look-
ing into the framework related posts and bringing out
the issues developers face.

3 METHODS

3.1 Data Extraction

The data dump of Stack Overflow contains all the data
of the website in XML form. We obtained the data
dump of August 2016. Among the XML files residing
in the dump, we used only the files relevant with our
purpose, namely posts.xml, tags.xml and users.xml.
We ran a script to extract these XML files to cor-
responding MySQL database tables. The database
schema of stack exchange files is available in this
link (Meta, 2017).

3.2 Data Processing

Having built our database, we focused on retriev-
ing the Laravel and Django related questions from
the Post table. The Post table contained a whopping
32,209,817 entries. Among these entries, 12,350,818
are questions. From these questions, we used a tag-
based search to extract 59,360 Laravel related ques-
tions and 130,588 Django related questions.

After that, we targeted on extracting the body of
each question and preparing them for textual analysis
using topic modeling. The bodies can contain HTML
tags and several unwanted strings. We trimmed out
the HTML tags and unwanted strings from the bodies.
Then, we created separate files for each body and kept
them in a directory. These files will represent a corpus
documents to be used in topic modeling.

3.3 Topic Modeling using LDA

Inference of topics from a corpus of documents is
performed by Latent Dirichlet Allocation (LDA). Ac-
cording to the LDA model, each document contains
a mixture of topics. Topics are also allowed to exist
across several documents. So it is easy to discover
themes and ideas that represents all the documents as
a whole (Rosen and Shihab, 2016).

To perform topic modeling, we used MALLET
version 2.0. It is a Java-based package for statis-
tical natural language processing, document classi-
fication, clustering, topic modeling, information ex-
traction, and other machine learning applications to
text (MALLET, 2017). The MALLET topic model-
ing tool uses LDA.

Running MALLET on a corpus requires two hy-
perparameters; namely: topic numbers and words per
topic. After assembling our data, we ran the MAL-
LET tool with different combinations of these hyper-
parameters. We performed the experiment with 20
topics and 10 words per topic, 20 topic and 20 words
per topic, 50 topics and 20 words per topic. After
each run, We manually checked the words in each
topic and through consensus, we concluded that the
result found with 50 topics and 20 words per topic
captures the theme of data better than the others. So
we retained 50 topics and 20 words per topic as our fi-
nal hyperparameters. The output is distributed among
two files. One, namely “key.txt”, contains word sets
of each topic along with their Dirichlet parameters.
The second file, namely “composition.txt”, contains
Having analyzed the words for each topic, we noticed that some of the word sets are similar in meaning. So we merged word sets conveying similar meaning and named the topics manually. Through this merging process, we obtained 18 topics for Laravel and 20 topics for Django. Thus we obtain the answer to our first research question, the issues users of these two frameworks face.

The naming of the topics was performed through consensus among the authors and the peers. A few samples of the naming and merging process have been shown in 1 and 2. The full list could not be tabulated here due to space constraints. The complete list can be viewed in (Authors, 2018b) and (Authors, 2018a). In both of the frameworks, we named a topic as "General Issue", which contains words which could not be classified to any specific topic. However, the human consensus is error prone. The naming and merging process of the topics may vary depending on the perspectives of developers. We are not claiming it to be anything concrete.

### 3.4 Finding Popular and Difficult Issues

With our topics identified by human understandable names, we focused on answering our research questions. To answer our first research question, we calculated the most dominant topics of each post with the help of the composition file generated by MALLET. Next, we calculated total questions and total views for each topics using these posts where that particular topic is dominant. Then we sorted the topics by view per question ratio. Naturally, the topic that has the most view per question is more popular.

To answer our third research questions, we determined the difficulty level of each topic by associating with it the mean and median duration in minutes of getting an accepted answer after a related question is posted. We ranked the topics by median time as according to Rosen et al. (Rosen and Shihab, 2016) the mean is likely to be skewed by long latency responses.

### 3.5 Finding Common Issues

For answering our final research questions, we manually checked the two sets of topics: The first set being the topics generated from Laravel (L) related posts and the second set being the topics generated from Django related posts (D). Using these sets, we obtained three other sets as follows:

- \( L - D \) (Issues only faced by users using Laravel Framework)
- \( L \cap D \) (Issues those are common for both users of Laravel and Django)
- \( D - L \) (Issues only faced by users using Django Framework)
Table 2: Set of words, their merging and given name of Django topics.

<table>
<thead>
<tr>
<th>Merged given name of set of words</th>
<th>Set of words</th>
</tr>
</thead>
<tbody>
<tr>
<td>file upload</td>
<td>file image upload images files uploaded django save download path video user photo uploading picture code media thumbnail image field uploads</td>
</tr>
<tr>
<td>form</td>
<td>form data fields field view user validation model forms.py submit save views.py formset post template input model form django create page button ajax html view click code jquery javascript function form user django template submit views.py link load display call form field select fields model widget list choices selected user values choice option input dropdown add set forms django box date time event day events model datetime month format dates year field days current calendar hours django set number start</td>
</tr>
<tr>
<td>migration</td>
<td>database django migration migrations run table south tables error created model migrate data app create models syncdb command manage.py running</td>
</tr>
<tr>
<td>authentication</td>
<td>file line return usr/local/lib/python c:python usr/lib/python lib/python error traceback exception init pid call kwargs import recent request response python tid email user password login facebook authentication username django users send account app token registration auth emails address oauth social google user users model profile create permissions group add created django permission custom access user profile username logged groups view user’s app user login page session logged view django redirect site users log cookie set authentication middleware url redirected sessions cookies access</td>
</tr>
</tbody>
</table>

4 FINDINGS

4.1 Evolution of Framework Related Posts

We looked into the trend of Laravel and Django related question in Stack Overflow quantitatively. First, we measured the percentage of Laravel related post to total post in each year. We measured the same for Django related post. The public data dump we used for topic modeling had was updated until August 2016. But for the trend of the Laravel and Django related post, we used the Stack Exchange Data Explorer (Exchange, 2017b) which enabled us to run queries online with the latest Stack Overflow data. The trend is presented in Figure 2.

4.2 RQ1

To answer our first research questions, we analyzed the word sets generated by mallets and manually provided understandable human names for the topics. The naming was done with consensus among peers proficient in these two frameworks. The topics for Laravel framework were named as Authentication, API, Date, Database, Documentation, Form, Image, Installation, General Issue, Language, MVC, Application, Session, Syntax, Transaction, Unit Testing, Elixir, and View. Similarly, the topics for Django framework were identified by the following names: Admin, API, App, Authentication, Cross-Site Request Forgery, Documentation, Encoding, File Upload, Form, General Issue, Installation, Localization, Migration, Model, Pep, Query, Server, Template, Unit Testing and View. Collectively, these are the issues that developers of these web frameworks face.

4.3 RQ2

Having named the topics, we attempted to rank them by popularity. For that purpose, we calculated the total number of questions and total views for each topic.

Figure 2: Percentage of related post per total posts per year.

Application, Session, Syntax, Transaction, Unit Testing, Elixir, and View. Similarly, the topics for Django framework were identified by the following names: Admin, API, App, Authentication, Cross-Site Request Forgery, Documentation, Encoding, File Upload, Form, General Issue, Installation, Localization, Migration, Model, Pep, Query, Server, Template, Unit Testing and View. Collectively, these are the issues that developers of these web frameworks face.

4.3 RQ2

Having named the topics, we attempted to rank them by popularity. For that purpose, we calculated the total number of questions and total views for each topic.
### Table 3: Popularity rank of Laravel and Django Issues.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Topic</th>
<th>Laravel</th>
<th>Django</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Installation</td>
<td>5459</td>
<td>1962</td>
</tr>
<tr>
<td>2</td>
<td>Documentation</td>
<td>1329</td>
<td>6610</td>
</tr>
<tr>
<td>3</td>
<td>General-Issue</td>
<td>10337</td>
<td>3633</td>
</tr>
<tr>
<td>4</td>
<td>MVC_basic</td>
<td>10766</td>
<td>5693</td>
</tr>
<tr>
<td>5</td>
<td>Database</td>
<td>9638</td>
<td>3117</td>
</tr>
<tr>
<td>6</td>
<td>Session</td>
<td>2207</td>
<td>3062</td>
</tr>
<tr>
<td>7</td>
<td>Syntax</td>
<td>3842</td>
<td>10207</td>
</tr>
<tr>
<td>8</td>
<td>date_time</td>
<td>680</td>
<td>12187</td>
</tr>
<tr>
<td>9</td>
<td>Form</td>
<td>3189</td>
<td>6922</td>
</tr>
<tr>
<td>10</td>
<td>Image</td>
<td>652</td>
<td>3119</td>
</tr>
<tr>
<td>11</td>
<td>Authentication</td>
<td>2838</td>
<td>3343</td>
</tr>
<tr>
<td>12</td>
<td>API</td>
<td>1118</td>
<td>11401</td>
</tr>
<tr>
<td>13</td>
<td>View</td>
<td>4305</td>
<td>11791</td>
</tr>
<tr>
<td>14</td>
<td>Application</td>
<td>1140</td>
<td>6488</td>
</tr>
<tr>
<td>15</td>
<td>language-translate</td>
<td>181</td>
<td>15615</td>
</tr>
<tr>
<td>16</td>
<td>Elixir</td>
<td>430</td>
<td>20068</td>
</tr>
<tr>
<td>17</td>
<td>unit_testing</td>
<td>448</td>
<td>9284</td>
</tr>
<tr>
<td>18</td>
<td>Transaction</td>
<td>801</td>
<td>775</td>
</tr>
<tr>
<td>19</td>
<td>unit_testing</td>
<td>430</td>
<td>1235</td>
</tr>
<tr>
<td>20</td>
<td>Language-translate</td>
<td>181</td>
<td>3003249</td>
</tr>
</tbody>
</table>

**CSRF** = Cross Site Request Forgery

### Table 4: Difficulty rank of Laravel and Django Issues on basis of time to get accepted answer.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Topic</th>
<th>Laravel</th>
<th>Django</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>unit_testing</td>
<td>16034.7</td>
<td>API</td>
</tr>
<tr>
<td>2</td>
<td>Elixir</td>
<td>7812.9</td>
<td>18763.1</td>
</tr>
<tr>
<td>3</td>
<td>Installation</td>
<td>13118.8</td>
<td>unit_testing</td>
</tr>
<tr>
<td>4</td>
<td>API</td>
<td>15978.9</td>
<td>Localization</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td>11696.4</td>
<td>Server</td>
</tr>
<tr>
<td>6</td>
<td>Transaction</td>
<td>45202.3</td>
<td>Migration</td>
</tr>
<tr>
<td>7</td>
<td>Application</td>
<td>10116.3</td>
<td>CSRF*</td>
</tr>
<tr>
<td>8</td>
<td>Authentication</td>
<td>8466.89</td>
<td>file-upload</td>
</tr>
<tr>
<td>9</td>
<td>View</td>
<td>6251.58</td>
<td>Admin</td>
</tr>
<tr>
<td>10</td>
<td>language-translate</td>
<td>5924.4</td>
<td>Authentication</td>
</tr>
<tr>
<td>11</td>
<td>Database</td>
<td>5504.18</td>
<td>App</td>
</tr>
<tr>
<td>12</td>
<td>date_time</td>
<td>3156.09</td>
<td>Documentation</td>
</tr>
<tr>
<td>13</td>
<td>Image</td>
<td>4565.55</td>
<td>General-Issue</td>
</tr>
<tr>
<td>14</td>
<td>General-Issue</td>
<td>7311.93</td>
<td>Model</td>
</tr>
<tr>
<td>15</td>
<td>Form</td>
<td>3314.9</td>
<td>Form</td>
</tr>
<tr>
<td>16</td>
<td>MVC_basic</td>
<td>6001.73</td>
<td>pep</td>
</tr>
<tr>
<td>17</td>
<td>Documentation</td>
<td>4898.51</td>
<td>Template</td>
</tr>
<tr>
<td>18</td>
<td>Syntax</td>
<td>6876.79</td>
<td>Query</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td>Encoding</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td>View</td>
</tr>
</tbody>
</table>

**CSRF** = Cross Site Request Forgery

The ratio of view count by the number of question for each topic has been taken as the ranking metric for popularity. Table 3 presents the popular topics for each framework, sorted from most popular to least popular. From this table, it is evident that installation is a major issue over both the frameworks. Installation is the first step of adopting a framework. Laravel has several installation alternatives which may get confusing for beginners. Django users also face trouble with installation. So it is no wonder that it is one of
the most popular issues for both the frameworks.

Among other issues of Laravel, we can infer from
the table that the framework is lacking in quality doc-
umentation or tutorial for new users. Moreover, the
changes between the version which are continuously
rolling for the betterment of the framework also create
confusion among the users. The beginners often find
Laravel framework as their first MVC pattern frame-
work and need to understand its underlying meaning
and implementation. So it is one of the most popular
topics with the highest number of question.

Similarly, we can infer that topics related to JSON
encoding, deployment and proper installment of a
server along with its possible errors and bugs are the
most popular topics related to Django framework.

4.4 RQ3

To answer our next research question, we attempted to
determine the difficulty level of the topics by associ-
ating with it the mean and median duration in minutes
of getting an accepted answer and ranking the topics
by median time. The topics for each of the frame-
work are presented in Table 4, sorted by their diffi-
culty measurements.

An interesting observation from the difficulty
ranking of the frameworks is the contrasting nature of
difficulty and popularity. More formally, Issues with
less popularity are difficult, which is somewhat intu-
iteive. For example, the top two difficult issues of Lar-
avel are Unit Testing and Elixir which are also among
the least three popular issues of Laravel. The similar
phenomenon can also be observed in case of Django.

Another important finding by observing the two
tables is that issues related to installation are popular
as well as difficult to answer. This somewhat indicates
that, regardless of any framework, users face prob-
lems while installation and the problems are also diffi-
cult to resolve. This denotes lack of well-documented
installation guideline or confusing installation process
for both the frameworks.

4.5 RQ4

Having observed all the issues for both of the frame-
works, we separated the issues in 3 columns as shown
in Table 5. We find that about 50% of the issues are
common in both frameworks and the other 50% spans
over the unique issues for individual frameworks.

5 CONCLUSION AND FUTURE
WORK

Stack Overflow is one of the largest question-answer
sites for the programmers and developers. We en-
deavored to bring insight into the web framework re-
lated question in Stack Overflow. We extracted bodies
from questions of Laravel and Django and used topic
modeling to find the topics. We ranked the topic ac-
cording to popularity and difficulty and determined
the common topics between these two. We found that
the most popular topics in Laravel are installation,
documentation, general issues, MVC basic, database,
session, syntax, date-time, form. The popular top-
ics in Django are encoding, installation, server, ad-
min, Cross-Site-Request-Forgery, migration, general
issues, app, and view. We found that the topics in two
frameworks are about 50% similar and both frame-
works have 50% of the topics as their unique topics.
In this way, we made a comparison between these two
frameworks that would help the developers to further
better their product and the developer community will
also be helped through establishing a statistical way to
find what web developers are talking about.

In future, we want to conduct an empirical study
involving developers from industry as a complemen-
tary of this study. Both Laravel and Django are
an open-source framework, and their source code is
available on Github for feedback, bug report, etc. We
want to analyze with Github statistic of the frame-
works’ repositories to find more about what the com-
unity is talking and contributing to these frame-
works. We want to extend our research to other frameworks. Determining these frameworks’ issues will enable us to conduct a comparative analysis, and we would be able to establish a list of common issues comprising almost all of the web frameworks. We would also like to conduct a study which will predict the activity level of the frameworks in future, e.g., how many users will keep using a framework or switch to others. The impact of the version change on the users can also be a significant study in this area.

REFERENCES


Authors (2018a). Django Set of Words. [Online; accessed May 2018].

Authors (2018b). Laravel Set of Words. [Online; accessed May 2018].


