Why is Developing Machine Learning Applications Challenging? A Study on Stack Overflow Posts

Moayad Alshangiti
Rochester Institute of Technology
Rochester, NY 14623, USA
mma4247@rit.edu

Hitesh Sapkota
Rochester Institute of Technology
Rochester, NY 14623, USA
hxs1943@rit.edu

Pradeep K. Murukannaiah
Rochester Institute of Technology
Rochester, NY 14623, USA
pkmvse@rit.edu

Xumin Liu
Rochester Institute of Technology
Rochester, NY 14623, USA
xmlics@rit.edu

Qi Yu
Rochester Institute of Technology
Rochester, NY 14623, USA
qi.yu@rit.edu

Abstract*—*Background: **As smart and automated applications pervade our lives, an increasing number of software developers are required to incorporate machine learning (ML) techniques into application development. However, acquiring the ML skill set can be nontrivial for software developers owing to both the breadth and depth of the ML domain.**

Aims: **We seek to understand the challenges developers face in the process of ML application development and offer insights to simplify the process. Despite its importance, there has been little research on this topic. A few existing studies on development challenges with ML are outdated, small scale, or they do no involve a representative set of developers.**

Method: **We conduct an empirical study of ML-related developer posts on Stack Overflow. We perform in-depth quantitative and qualitative analyses focusing on a series of research questions related to the challenges of developing ML applications and the directions to address them.**

Results: **Our findings include: (1) ML questions suffer from a much higher percentage of unanswered questions on Stack Overflow than other domains; (2) there is a lack of ML experts in the Stack Overflow QA community; (3) the data preprocessing and model deployment phases are where most of the challenges lay; and (4) addressing most of these challenges require more ML implementation knowledge than ML conceptual knowledge.**

Conclusions: **Our findings suggest that most challenges are under the data preparation and model deployment phases, i.e., early and late stages. Also, the implementation aspect of ML shows much higher difficulty level among developers than the conceptual aspect.**

***Index Terms*—Machine Learning, Software Development, Stack Overflow, Data Mining.**

I. Introduction

With a rapid development of machine learning (ML) technologies, knowledge discovery from large-scale data has attracted significant interest from various application domains. Commonly used ML toolkits, such as Scikit-Learn and TensorFlow, have provided a large number of ML libraries that allow software developers to programmatically integrate data analytic functionalities into their software applications. Also, there is a growing demand for industry adoption of ML.

978-1-7281-2968-6/19/$31.00 ©2019 IEEE

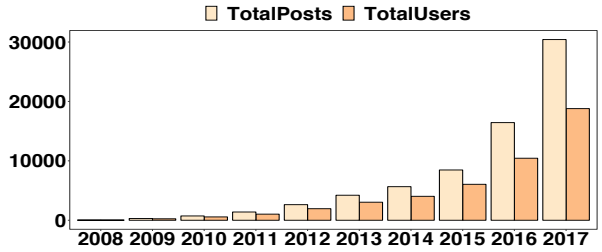


Fig. 1. Number of machine learning related question posts overtime

According to the most recent NVP Big Data and AI Executive Summary [1], 92% of organizations are increasing their pace of investment and 62% of them have already seen measurable results from their investments in big data and AI.

The tremendous increase in ML adoption poses new challenges for software developers. Our analysis shows that the number of ML-related questions and users in Stack Overflow is roughly doubling every year over past ten years as shown in Figure 1. This indicates that not only the amount of developer interest in using ML techniques is increasing (especially after 2015), but also that software developers face various challenges in ML related development tasks.

Besides facing some common challenges of using new software libraries (e.g., installation, configuration, and debugging), a developer may face some unique development issues due to the interdisciplinary nature of ML domain. The proper usage of ML libraries and interpretation of the results require not only programming knowledge, but also a certain level of expertise in related fields such as statistics, linear algebra, and visualization. For example, Scikit-Learn provides a number of libraries for dimensionality reduction, e.g., subset selection, l_1-regularization, and principle component analysis (PCA). However, each library comes up with a number of parameters that need to be tuned based on the dataset and the development task in order to optimize the performance, such as stopping criteria and penalty on model complexity [2]. The description of the generated models can be quite comprehensive, including those related to coefficients, prediction result, prediction

confidence, and so on. Therefore, choosing the appropriate library, properly using it, and interpreting and handling the result would require deep knowledge of those libraries as well as the ML theories.

Software developers, regardless of their programming background, could easily get stuck on ML steps if they do not have sufficient training on the corresponding topics. Also, the major steps in an ML task, such as preparing data, modeling and selecting features, building and utilizing models, and visualizing and analyzing results are interrelated. For example, failing to normalize data might cause issues later in dimensionality reduction models, such as PCA. That means the improper usage of libraries or implementation of one step could affect the functioning of a different step for that task. As a result, the incorrect or unexpected output, caused during the earlier steps, can be only revealed at a later step. It would require software developers to have deep understanding of those steps and the selected libraries to identify such causal relationship among steps and properly fix the errors.

Despite its importance and urgency, there is little work on systematically understanding the challenges involved in ML application development. Patel et al. [3], [4] conducted an early (2008) study on the difficulties developers face in creating an ML component of an application, finding the key difficulties as (1) following the iterative and exploratory process of ML, (2) understanding data and output, and (3) evaluating the performance of ML techniques in the context of specific applications. However, a decade has elapsed since this study, and since then, as we described earlier, there has been a tremendous growth in the variety of ML techniques, application domains, and the number of developers using ML.

We seek to understand the challenges software developers face in engineering ML applications. However, unlike Patel et al.'s laboratory study (which involved interviews of researchers and graduate students creating one specific application), we study tens of thousands ML-related posts from a variety of developers on Stack Overflow (which is one of the most popular Q&A forums for software developers and has been used to answer a variety of software engineering research questions, e.g., [5], [6], [7]). Thus, our study has a broader coverage and also sheds light on contemporary challenges.

Our study asks five research questions (RQs). The analyses we perform to answer these RQs can provide an in-depth overview on the development challenges as well as insights on how to assist the developers in addressing those challenges.

RQ_1: Are machine learning questions more challenging to answer than other questions on Stack Overflow?

RQ_2: Do we have enough experts on Stack Overflow to address machine learning questions?

RQ_3: What phases of machine learning are the most challenging for software developers?

RQ_4: What are the most popular and most challenging machine learning topics?

RQ_5: What kind of knowledge is needed to address developers' challenges with the use of machine learning?

II. Data Preparation

In our study, we analyze data from Stack Overflow, a Stack Exchange website dedicated for "professional and enthusiast programmers." Stack Overflow is, by far, the largest of the Stack Exchange websites (in terms of the overall data dump size) and has been used for many SE related research [8]–[14]. Stack Overflow's topic (programming) and popularity makes it an ideal data source for studying the challenges involved in developing ML applications. Specifically, we employ the (question and response) posts (65,376,980), comments (66,432,641), and users (3,823,800) data from the data dump (the full data dump also includes other information such as Post History, which we do not analyze in this study). The data dump covers July 2008–June 2018 period [1].

A. Identifying Machine Learning Related Questions

As a first step towards answering our research questions, we identify the subset of Stack Overflow questions that capture developers' ML challenges. To do so, we used a snowball sampling approach where we started with the "machine-learning" tag and expanded the list based on co-occurrence and relevance. For each level we only inspected the top 25 tags. For example, at the first level, we went through the list of top 25 tags that co-occurred with the tag "machine-learning" and added those that are exclusively used with ML, e.g., although the tag "java" co-occured with other tags, it was not added because it is not exclusively used with ML. We repeated this process until we complied the list of 50 ML tags shown in Table I. As we can observe from the table, our list captures a wide range of ML topics, covering problems (e.g., image-recognition), concepts (e.g., supervised-learning), models (e.g., SVM), and libraries (e.g., scikit-learn).

B. Study Sample and Data Annotation

To investigate both the quantitative and qualitative aspects our research questions and compare it against a baseline, we created three samples (available online [2]):

1) Quantitative Study Sample: we created a quantitative sample of the Stack Overflow data dump that consists of all the *question* posts tagged with at least one tag from Table I, along with all their *response* posts and *comments*. We also included the *users* posting the questions and responses (and removed 492 questions with deleted user accounts). The final quantitative sample consisted of 86,983 question posts generated by 50,630 unique users. The questions in this sample cover the period from July 2008 to June 2018.

2) Qualitative Study Sample: In some of our analyses (e.g., in answering RQ_3 and RQ_4), we needed a qualitative analysis of manually labelled data. Examples of such analyses include (1) identifying the challenging phases of ML for developers (e.g., are most challenges under model training or model evaluation?), (2) identifying the type of background knowledge required to address an ML question (e.g., is it more common

[1] https://archive.org/details/stackexchange

[2] https://github.com/mshangiti/esem2019

TABLE I
TOP 50 MOST COMMONLY USED MACHINE LEARNING TAGS

Tag	Freq.	Tag	Freq.
machine-learning	24616	neural-network	13216
supervised-learning	326	conv-neural-network	2445
unsupervised-learning	281	recurrent-neural-network	1067
reinforcement-learning	617	rnn	621
prediction	1657	deep-learning	8296
regression	8555	image-recognition	941
linear-regression	2811	object-detection	1444
non-linear-regression	300	sentiment-analysis	1032
nls	363	cluster-analysis	3654
classification	5679	hierarchical-clustering	676
classifier	429	pca	1473
multilabel-classification	276	autoencoder	417
multiclass-classification	148	word2vec	1056
document-classification	207	word-embedding	280
text-classification	821	tf-idf	797
logistic-regression	1626	rfe	59
svm	3356	feature-engineering	41
svmlight	96	feature-selection	701
decision-tree	1380	feature-extraction	969
random-forest	1695	cross-validation	1152
naivebayes	625	confusion-matrix	342
perceptron	327	precision-recall	201
dbscan	265	scikit-learn	11285
knn	831	tensorflow	28360
k-means	1960	r-caret	1137

to see challenges with ML implementation or concepts?), and (3) verifying whether questions with no accepted answer are truly not answered (e.g., an answer may exist but the asker did not mark it as accepted).

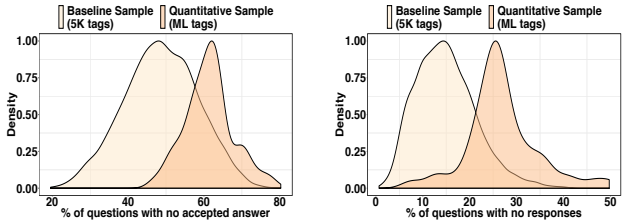
Since it is not feasible to manually annotate all 86,983 ML questions, we instead created a qualitative sample to address these types of questions. Specifically, we randomly picked 50 users from the quantitative sample while making sure that each selected user has asked at least ten ML-related questions. Then, we compiled a list of all ML-related questions from each of the 50 users (which led to a total of 684 questions).

Two PhD students (in computing) annotated the qualitative study sample separately and then compared their labels. Disagreements were resolved in discussions involving all team members. We will discuss the specifics of annotation tasks under the corresponding research question in Section III. Since we labelled data along multiple dimensions and no previous guidelines (e.g., codes or taxonomies) were available, the labelling process was both challenging and time consuming.

3) Baseline Sample: To answer some of our questions (i.e, RQ_1 and RQ_2), we need a baseline sample to compare our results against. For example, to compare the difficulty level of ML questions against other domains, the number of experts in the Stack Overflow ML community against other domains, etc. In order to construct a baseline sample capturing the common overall trends in Stack Overflow data dump as a whole, we picked a sample that includes all the questions under 5000 tags (10% of all existing Stack Overflow tags), covering a wide range of domains such as programming languages (e.g., Java), protocols (e.g., HTTP), databases (e.g., MySQL), HCI (e.g., user-interface), services (e.g., google-maps), and so on.

III. APPROACH AND RESULTS

In this section, we present the investigation on the proposed research questions. For each question, we describe its motivation, the approach used to explore the answer, and the findings.



(a) No Accepted Answer (b) No Response

Fig. 2. Are ML-related questions more challenging to answer?

RQ_1: Are ML questions more challenging to answer than other questions on Stack Overflow?

Motivation: Given the significant growth of ML questions posted on Stack Overflow (Figure 1), we start with studying whether ML questions are actually considered more challenging than other questions on Stack Overflow.

Approach: We measure the challenge or difficulty of a question using the indicators adopted in previous similar studies [15]–[18]. The indicators include the percentage of questions with no accepted answers, the percentage of questions with no response at all, and the average response time taken to receive an accepted answer. We use median response time instead of mean to reduce the impact of outliers. To avoid our study to be affected by those questions posted too recent to collect information for the indicators, we only consider those questions posted for at least six months before. To conduct the comparison for the first two indicators, we compare ML questions in the quantitative sample against questions in our baseline which captures the general Stack Overflow's trend found in other domains. For the third metric (i.e., average response time), we select web development, the most popular domain on Stack Overflow as reported in [11], as to specific domain to compare ML questions to. We randomly selected a subset of web development questions from the baseline sample (similar in size to the quantitative sample) for the comparison.

Results: Figure 2(a) shows the distribution of questions with no accepted answers under machine learning tags against the baseline sample. The Figure clearly indicates that the ML distribution has a much higher mean of 61%, which indicates that the majority of ML tags suffer from a higher percentage of questions with no accepted answers compared to our baseline distribution with a mean of 48%. This observation itself is not sufficient to indicate that ML application development is more challenging than in other domains. Users are not always responsible enough to mark a reasonable answer as accepted. Not having an accepted answer does not necessarily mean the question is too challenging to answer. Therefore, we also compared the percentage of questions with no responses at all with the baseline in Figure 2(b). Similarly, we can visually observe that the ML distribution has a much higher mean

than the baseline, which indicates a much higher percentage of questions with no responses can be found under ML than the general trend of other domains on Stack Overflow.

Observation 1

Machine learning questions suffer from a higher percentage of questions with *no accepted answer* than other domains on Stack Overflow.

Observation 2

ML questions show a higher percentage of questions with *no response* than other domains on Stack Overflow.

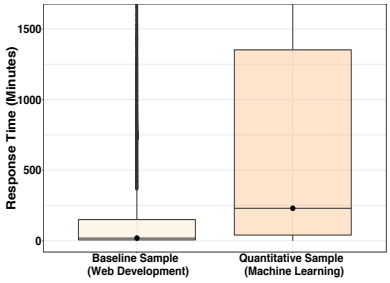


Fig. 3. Average median time to receive an answer in ML vs Web Dev.

In Figure 3, we present the response time for the quantitative sample and the web development subset sampled from our baseline. We can visually observe that web development questions are mostly concentrated below 150 minutes response time, whereas, ML questions have a much wider spread. In fact, we found that the median average time for ML questions to receive an answer is 230 minutes (roughly 4 hours), whereas, the median average time for web development questions is 19 minutes. The difference is very significant and clearly indicates how much more challenging it is to answer ML related questions compared to the questions of a commonly seen domain on Stack Overflow such as web development. Moreover, Asaduzzaman et al. [19] conducted an analysis on roughly 1.3 million Stack Overflow questions and reported an average of 15 minutes response time, which shows that the short response time (below 20 minutes) is not only for the web development domain, but rather is the common norm for Stack Overflow.

Observation 3

On average, a ML question takes ten times longer to be answered than the typical Stack Overflow question.

RQ_2: Do we have enough experts on Stack Overflow to address machine learning questions?

Motivation: Asaduzzaman et al. [19] reported that the leading cause behind the unanswered questions on Stack Overflow is the failure to attract an expert, due to several reasons, one of which is the lack of experts in the community. In this research question, we want to investigate whether a lack of active experts is one of the factors behind the difficulty in getting answers to ML questions.

Approach: Our first step is to establish a baseline to compare ML domain with. We select web development domain for the baseline due to its popularity on Stack Overflow. We sample a web development set, which has the same number of users as our quantitative sample (Approx. 50,630 users). Given the users from the two domains, we performed two types of comparisons. The first comparison utilized the ExpertiseRank approach [20], which finds experts within online communities, and has been demonstrated to perform well on question-answering communities. ExpertiseRank creates a directed graph (where nodes are users) that captures the relationship between the users within a given community. It assigns an expertise score to a user, considering both how many other users the user has helped and also whom the user has helped. For example, if a user B is able to a answer user A's question, then user B's expertise rank will be higher than user A. Also, if user C is able to answer B's question, then user C's expertise rank should be higher than both user A and B not only because he/she was able to answer user B's question, but also because he/she specifically helped user B who has demonstrated expertise by answering user A's question. Using the ExpertiseRank approach, we compared the distribution of expertise scores for the ML domain with the distribution of the web development domain.

For the second comparison, we employed manual annotation. That is, first, we randomly selected 50 users from each of the sample domains. Next, two annotators labelled each user in each sample as *novice*, *intermediate*, or *expert* with respect to the corresponding domain. To label the expertise level of a user, the annotators looked for evidence in the user's bio profile, in the list of tags that user is most active under, and in the corresponding tag score, which is a community based score that measures the overall value of the user's questions and/or answers (total upvotes minus total downvotes).

Results: Figure 4 compares the ExpertiseRank distributions of ML against web-development users. Since the ExpertiseRank values are quite low (they add up to one for each sample), we applied a logarithmic transformation on the expertise score to better visualize the distributions. From this figure, we can observe two long tail distributions, each with two observed masses. The first is the left most mass, where the majority of users are concentrated, should represent the novice users, i.e., the ones with the lowest expertise score. The second and more interesting mass is the right most one which should represent the expert users, i.e., the ones with the highest expertise rank. We can observe a much higher peak for the web development domain and a much smaller peak for the ML domain, which confirms that ML indeed has a much lower number of experts than the web development domain.

For our manual comparison, using Cohens Kappa we found a nearly perfect agreement level (kappa=0.92). Figure 5 shows the result of the comparison where we can observe that the ML domain has a much smaller percentage of experts and intermediate users compared to the web development domain. Whereas, the percentage of novice users is much higher in ML than web development.

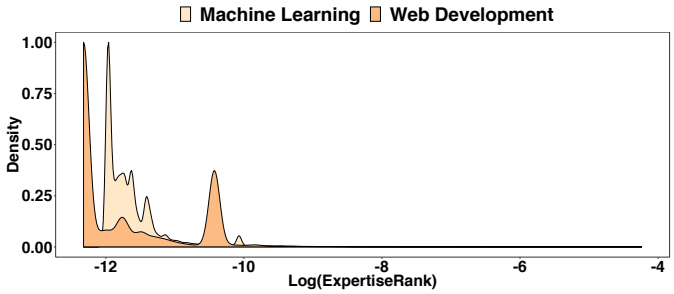


Fig. 4. Distribution of experts in machine learning versus web development.

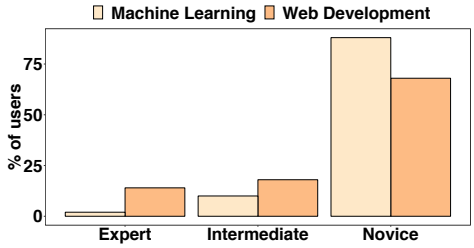


Fig. 5. Percentage of experts in machine learning versus web development.

Observation 4

There is a lack of ML experts on Stack Overflow.

RQ$_3$: What phases of machine learning are the most challenging for software developers?

Motivation: To develop ML applications, a software developer may need to go through a complete ML life cycle starting from the problem formulation, to model creation, and ending with model deployment. In this research question, we aim to investigate what phase/step of ML is considered the most challenging.

Approach: We use the complete qualitative study sample (684 questions) and label each question as per the labels in Table II. The goal is to study the distribution of challenges across different ML phases.

Results: We found a substantial agreement (kappa=0.8) between the two annotators for the qualitative analysis shown in Figure 6. We observed that developers face difficulties across all ML phases. However, the *data pre-processing and manipulation* phase (DP) and the *model deployment and environment setup* phase (MD) show that roughly 50% of the questions with no accepted answer fall under them. Having the MD phase as the most challenging phase may indicate a difficulty in transferring the knowledge from the example demos learned from online tutorials and courses to a specific problem. As for the DP phase, it seems to be the second most challenging as many users may lack the programming skills needed for data slicing and dicing tasks, such as those with insufficient computing background. Also, we observed that difficulty in this phase comes from the challenges of dealing with unstructured data, such as how to properly convert text or images into features that can be used in an ML model.

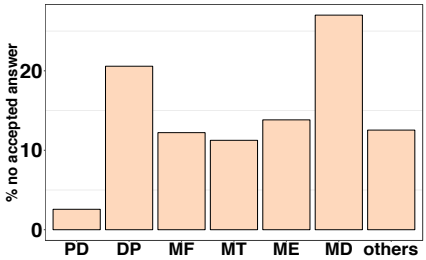


Fig. 6. Percentage of ML questions under each ML phase.

Also, we observed that the percentages of questions with no accepted answer among MF, MT, and ME phases are quite high and similar. We believe the difficulty with these phases is due to developers jumping directly into creating demos without spending enough time to digest the required ML concepts, which can lead them to a lot of confusion around training/testing error, regularization, hyper-parameter tuning, convergence determination, and so on. Table III shows several examples of some confusions we observed on basic ML concepts. Finally, the *others* category consists of those questions that are difficult to be placed under a phase. Those questions can be further broken down into two types: (1) questions that represent a general machine learning conceptual question (MLC), e.g., "*how does k-means clustering work?*" (2) questions that represent a general machine learning library inquiry, e.g., "*what is the difference between fit and fit_predict*". We observed that the majority of questions under others belong to the second type, which indicates a lack of proper documentation for specific methods and/or parameters as mostly are questions on the proper use of a specific method and/or on the difference between two specific methods or parameters for a given library.

Observation 5

The challenges developers face span across all ML phases. However, the DP and MD phases are the most challenging.

RQ$_4$: What are the most popular and most challenging machine learning topics?

Motivation: Compared to RQ$_3$, this question focuses on investigating the ML application development challenges in a more fine-grained manner, i.e., identifying those popular and challenging ML topics. Such knowledge can provide more in-depth, and specific insights on the trends of ML adoption, the challenges, and the demands on supporting efforts.

Approach: We first identify the list of discussed topics in our quantitative study sample. We then investigate the popularity of those topics using their proportions and/or total number of views. We measure how challenging a topic is using the indicators introduced in RQ_1, e.g., the percentage of the questions with no accepted answer. The intuition is that topics with more unanswered questions and/or with longer response time are considered more challenging.

To identify the list of discussed topics in our quantitative study sample, we used the traditional topic modeling technique, Latent Dirichlet Allocation (LDA) [21]. LDA been widely used in many previous studies [11], [16], [22] to

TABLE II
LABEL DEFINITION FOR MACHINE LEARNING PHASES. THE PROVIDED *QuestionID* CAN BE USED TO VIEW THE ORIGINAL QUESTION ON STACK OVERFLOW USING THE LINK HTTPS://STACKOVERFLOW.COM/QUESTIONS/QUESTIONID

Label	Description	Example
Problem Definition and Exploration (PD)	We assume the developer has a business idea or a problem at this stage. Questions include anything on how to formulate a Machine Learning solution to solve a problem. Also, Questions related to better understanding of the data or the given problem before any modeling (exploration).	I have a task of prognosing the quickness of selling goods ... for example... the client inputs the price that he wants his item to be sold and the algorithm should displays that it will be sold with the inputed price for n days. And it should have 3 intervals of quick, medium and long sell... how exactly should I prepare the algorithm? ... My suggestion: use clustering technics for understanding this three price ranges and then solving regression task... Is it a right concept to do? (QuestionID# 39207524)
Data Pre-processing and Manipulation (DP)	We assume the developer is preparing his data for a ML model(s). Questions about data loading, data accessing, data cleaning, data splitting, data format changing, data labelling, data imbalance issues, data normalization, etc.	I have a dataset with medical text data and I apply tf-idf vectorizer on them ... my question ... TfidfVectorizer ... splits the text in distinct words for example: "pain", "headache", "nausea" and so on. How can I get the words combination ... "severe pain", "cluster headache", "nausea vomiting". Thanks (QuestionID# 45690619)
Model Fitting (MF)	We assume the developer has a specific model in mind (e.g., SVM), so questions related to a specific model implementation, training, convergence determination, etc.	My algorithm... linear regression stochastic update ... training error is more than testing... why? (QuestionID# 14616900)
Model Tuning (MT)	We assume the developer has trained a specific model and is aiming to fine tune it through hyper-parameter tuning, learning rate, regularization, etc.	I implemented a POS tagger using RNN. There are 3 features ... but the accuracy is very low... Just wondering if anyone know POStagger using RNN ... in python that I can compare it with my model ... because this model is not working. (QuestionID# 36938556)
Model Evaluation and Result Interpretation (ME)	We assume the developer completed the training and tuning of a single or multiple ML models. Questions related to evaluation or measuring the performance of a model. Questions related to results interpretation	I have a confusion matrix, now I need to calculate the precision, recall and FScore ... how do I do that using the obtained values? (QuestionID# 33463492)
Model Deployment and Environment Setup (MD)	Questions related to environment setup, memory or storage issues, deployment performance tuning, etc	I used to use theano for deep learning development... In Theano, it supports ... to store input data on GPU memory... is it possible ... for TensorFlow? (QuestionID# 37596333)
Others	Bucket for questions where the Machine Learning phase is not clear.	Difference between tf.nn.batch_normalization and tf.nn.fused_batch_norm? ... Is there a difference in what those two functions implement? (QuestionID# 43999951)

TABLE III
EXAMPLES WHERE DEVELOPERS ARE NOT SPENDING ENOUGH TIME TO DIGEST MACHINE LEARNING.

QID	Question Description	Issue
40216872	PCA for features selection.. assume.. dataset with 15 features..I want... the 5 most important features.	PCA is used for dimensionality reduction not feature selection
41697617	I'm creating a binary classifier .. How can I understand the importance of distinct value within a column (x1, days of week) for target variable?	Feature importance works on feature level. It cannot be calculated per observation.

summarize the topics of a large document corpus. The intuition behind LDA is that it leverages the textual content of a set of documents to group together the frequently co-occurring words into an approximation of a real-world concept, i.e., a topic. We created a ML corpus using a bag-of-words representation (including bigrams) generated from the textual content of the questions (i.e., title and body) in our quantitative study sample (86,983 ML questions). We then applied LDA to discover the discussed topics within the corpus. When training LDA, it is required to specify the estimated number of topics k beforehand, which determines the modeling result. We leverage two factors to determine the optimal k value: the model's perplexity and the generated topics coherence. A model's perplexity is a commonly used measure to evaluate the goodness-of-fit of a probabilistic model [21]. The lower the perplexity of the model on a held-out data, the better the stability and goodness-of-fit. The generated topics coherence is also important based on the analysis reported in [23] where they found that what humans observed as semantically related set of terms (representing a topic) does not always correlate with a high predictive likelihood (or equivalently perplexity). Thus, the topic coherence measurement Cv [24] describes the level of semantic correlation between the terms under each topic. The higher the coherence, the better the model.

With the best number of topics determined, we investigate the nature and meaning of each LDA's topic. First, we use the distribution of topic over words. In LDA, each word in the corpus dictionary is assigned a probability of belonging to a specific topic. The top terms with the highest probability can be used to infer what the topic is all about. As such, we look into the top terms of each topic to better understand the nature of the discussed topic. Moreover, each document, i.e., question post, is assigned a probability that describes how likely that

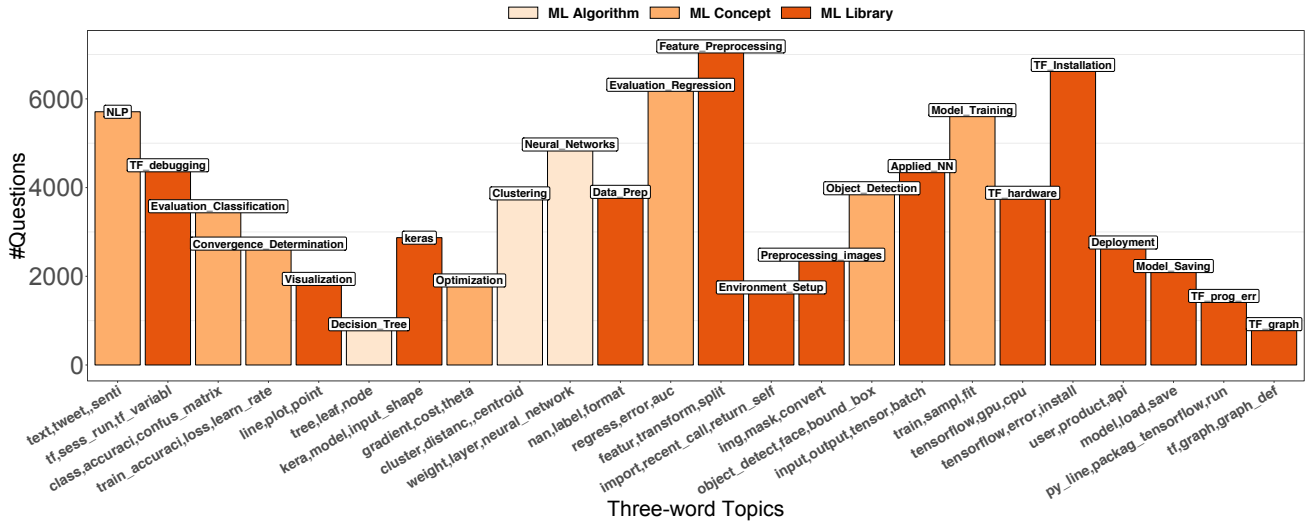


Fig. 7. The discovered LDA topics.

the post belongs to a given topic. We use this probability to infer the most dominating topic for a given question [16]. This allows us to further understand the type of questions that fall under a specific topic A, i.e., their difficulty and popularity.

Results: For the quantitative analysis, we found that the perplexity was the lowest and the topic coherence was the highest when the number of topics is 30. Also, through manual investigation, we found that most of the topics do correlate with a ML topic. However, a few of the topics were capturing the commonly used language in the corpus more than capturing a specific concept, i.e., can be considered noise or topics of stop words. As such, we are not reporting those topics in our summary in Figure 7 where we report the proportions of the topics (y-axis), the selected three words from the set of words with the highest probability of belonging to the topic (x-axis), the topic's label (top of the bar), and the general category of the topic. For example, the first topic is the dominant topic in roughly 6k questions, the words (text, tweet, sentiment) are the most representative words for the topic (from the set of top words), the label we assigned to the topic (Natural Language Processing or *NLP*), and generally we believe it falls under ML concepts.

We found that the discovered topics represent questions on a specific ML concept, ML algorithm, or ML library. We provide an example of each type in Table IV. For example, topics on ML concepts include questions on Natural Language Processing (NLP), evaluation of regression and classification models, convergence determination, optimization, object detection, etc. Whereas, topics on ML algorithms include questions on a specific ML algorithm such as decision trees, Neural Networks, and clustering models. Topics on ML libraries are questions on data manipulation (e.g., using numpy or pandas), data visualization (e.g. ggplot2), and ML algorithm libraries (e.g. Scikit learn). Figure 8 shows the most popular and most challenging ML topics. We can observe that the most challenging topics for software developers (right side of the Figure) are all related to environment setup and model deployment (e.g., environment setup, TF installation and hardware, model saving, etc). We can also observe that neural networks, TensorFlow, object detection, and NLP are quite hot topics (top side of the Figure). This could be caused by the rise in the popularity of deep learning techniques for object detection and NLP tasks.

TABLE IV
EXAMPLES OF THE THREE BROAD CATEGORIES OF LDA TOPICS

QID	Question Description	Topic
21618478	How do I choose between tf-idf document similarity and naive Bayes classifier....which one to use...	NLP (ML Concept)
42966393	I am training my method. I got the result as below. Is it a good learning rate? If not, is it high or low?	Convergence determination (ML Concept)
34997134	Relation between .. number .. trees and .. tree depth? .. tree depth should be smaller than the number of trees?	Decision Tree (ML Algo.)
1793532	How do I determine k when using k-means clustering?	Clustering (ML Algo.)
39395198	How to configure TensorFlow to use all CPU's	TF Hardware (ML Lib.)
48212028	How to save .. models so that it can be used later .. in RStudio?	Model Saving (ML Lib.)

Observation 6

The most challenging ML topics show difficulty with data and feature preprocessing, environment setup, and model deployment.

Observation 7

Neural networks and deep learning related topics and libraries are the most popular on Stack Overflow, especially for object detection and NLP tasks.

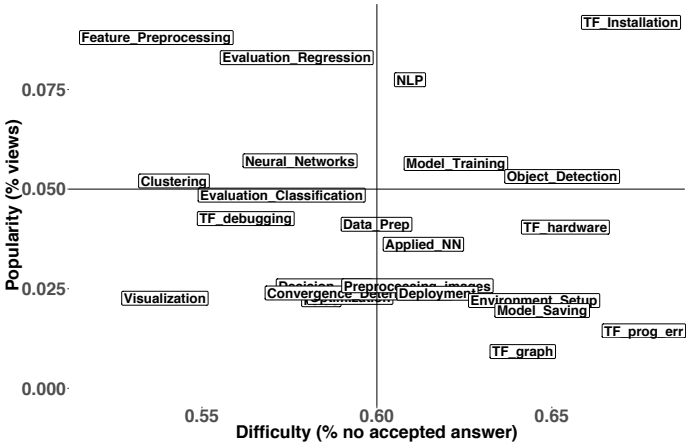


Fig. 8. The most popular and most difficult ML topics

Observation 8

The TensorFlow library has been widely adopted among developers but a high level of difficulty has been observed with the library's usage and environment setup.

RQ$_5$: What kind of knowledge is needed to address developers' challenges with the use of ML?

Motivation: The nature of knowledge required to answer an ML question can be conceptual (e.g., statistical details on how a model works) and/or implementation related (e.g., data slicing dicing, model construction, and deployment). To address questions that require conceptual knowledge, a solid understanding of ML concepts is needed, e.g., how specific models work, how their hyper-parameters tuned, etc. Whereas, questions that require implementation knowledge, an experience with ML model implementation is needed, e.g., experience with the use/troubleshooting of ML libraries and their underlying assumptions. In this question, which of these two types of knowledge is required to answer developers' ML-related questions on Stack Overflow.

Approach: We manually label the qualitative study sample (684 questions) under one of the categories listed in Table V. Our goal is to understand the type of background knowledge needed to answer ML questions.

Results: We found a substantial agreement level (kappa=0.78) between the annotators for the qualitative analysis in Table V. We observed that 66% of questions can be addressed with only a solid ML implementation knowledge. We found that many of the questions are due to misuse or misunderstanding of the used ML library. It's also common to see questions on how-to-do a specific machine learning task using a specific library. This high level of difficulty with ML implementation further supports our earlier theory that a good percentage of developers are facing trouble transferring their demo code from online courses or tutorials to their specific problem, which can explain the high percentage of questions that require implementation knowledge.

Observation 9

The majority of questions on Stack Overflow can be addressed with ML implementation knowledge.

We expected that the number of questions requiring ML conceptual knowledge and implementation knowledge would be similar. However, questions that require conceptual knowledge in ML constituted only 31% of our qualitative sample. This observation also supports our earlier conjecture that developers might be jumping too quickly to model implementation without spending enough time on understanding the underlying machine learning concepts.

Observation 10

Questions that require machine learning concept knowledge are less common than expected.

Finally, we observed that 71% of the questions requiring both conceptual and implementation knowledge do not have an accepted answer. This is expected given that such questions are more challenging and that there is a lack of experts in general in the community. However, this type of questions only represented 3% of our qualitative study sample.

Observation 11

Questions that require both ML conceptual and implementation knowledge suffer the most from having no accepted answer.

IV. Discussion and Implications

In this section, we will discuss our findings and their implications on the software engineering community.

A. ML requires a wide set of skills from different domains

We observed that one of the most challenging aspects of mastering ML is the fact that it requires a wide set of skills from different domains. For example, the problem formulation phase requires a good overall understanding of how ML techniques work to help transform the problem in hand into an appropriate ML learning task. Second, the data preprocessing and manipulation phase requires a good data slicing and dicing skills to do proper data cleaning and feature preprocessing e.g., normalization, scaling, etc). Third, the model fitting and tuning requires a solid understanding of the mathematical and statistical intuition behind the models (e.g., understanding regularization, optimization, etc). Treating ML as a blackbox can lead to very troubling confusions such as the ones reported in Table III. The model evaluation and result interpretation phase requires an understanding of the different measures, e.g., regression versus classification, and effect of using different datasets, e.g., having a balanced dataset versus an imbalanced dataset with rare classes. The model deployment phase requires skills in web services creation, system administration, distributed systems, and multi-threaded programming.

Thus, we believe ML is more challenging because it requires such a wide set of skills, which we observed in the data and reported many examples of in Tables II, V, and III.

TABLE V
LABEL DEFINITION FOR BACKGROUND KNOWLEDGE TYPE NEEDED.

Label	Description	Example	QID
Conceptual knowledge	The question requires an understanding of one or more machine learning concept(s). The question is mainly about a machine learning algorithm/model. This includes what machine learning model to use for a specific problem, how the models work, the meaning of their parameters, the math connection behind them, etc.	Which topology is correct for segmentation? I have an image size of WxHx3 that needs to segment into 21 classes...Which topology is better? I have seen both of them. No.1 in FCN (fully convolution network for semantic segmentation), No.2 in VoxResNet, UNet.	43555221
		What should we try: low or high learning rate?	45427620
		In the LSTM architecture, there are two kind of activation functions, tanh and sigmoid.. I want to derive the BPTT algorithm for the unfolded LSTM network just like RNN..	50511168
Implementation knowledge	The question can be answered through the documentation of a specific library or through users with experience with the library's troubleshooting or implementation. E.g., issues with using the wrong function, providing the wrong parameters, etc.	I want to save the current trained model. and load the saved model later ... When saving, should I save the classifier, or theano functions?...	31612074
		Caffe2's equivalent to Caffe's Net.backward()?	48511018
		Im trying to train a Keras model based on partial features but when I'm calling the fit method...I get the following error...	45479239
Both	The question requires knowledge in both Conceptual and Implementation knowledge.	Different learning rate affect to batchnorm setting. Why? I am using BatchNorm layer... In solver.prototxt, I used the Adam method. I found an interesting problem that happens in my case. ... Could you suggest any reason for that? Thanks all	44242122

B. Identifying the most challenging ML topics, phases, and the background knowledge needed to address them

We demonstrated using quantitative and qualitative analysis that most of the developers challenges with ML are under data preprocessing and manipulation (DP), environment setup, and model deployment (MD), i.e., the first and the last phases of ML. The DP and MD phases are mostly overlooked and not given proper attention, which may explain the higher percentage of difficulty. Thus, companies working on ML should aim to provide more support to junior developers with environment setup and model deployment. We also showed a high popularity and difficulty with neural networks and deep learning related techniques and libraries. This might be due to the wide adoption of deep learning in the last few years. As such, providing more online tutorial and educational content on those topics can greatly help address their questions.

We highlighted that the majority of challenges can be addressed using ML implementation knowledge. On the one hand, this high percentage highlights the need for a better detailed documentation of machine learning libraries. These include adding a better explanation for the methods and parameters, adding more tutorials that show the proper usage with examples covering different user cases, and possibly adding a FAQ section that addresses the common questions found on Stack Overflow or other similar community based platforms. On the other hand, increasing the size of the documentation may make finding the appropriate reference difficult. Thus, an automated system that recommends appropriate documentation reference might help address many of the challenges found in those questions.

Moreover, using Figure 8, developers trying to learn ML can begin with more popular and less challenging topics such as *regression models* as it can be easier to find answers under such topics. Also, developers working for ML libraries such as TensorFlow can use the provided insight to further investigate the reasons behind the high difficulty developers face with their library. Finally, ML educators can put more teaching effort on topics that are more difficult to the community such as *object detection*, and less effort on topics that are less popular and less challenging such as *data visualization*.

V. THREATS TO VALIDITY

We explored five RQs on developer challenges with ML, and offered our results as elevent observations. However, there are several threats to the validity of these results.

First, our study is focused solely on Stack Overflow posts. Accordingly, our conclusions are limited to the types of ML challenges developers discuss on Stack Overflow. Additional studies must be conducted to generalize these findings to the broader challenges of ML application development. Second, to create the quantitative sample, we stopped at 25 tags, which does not provide a complete sample of all ML related questions, but we believe it provides a sample with enough statistical significance. Third, in answering RQ_3 and RQ_5, we employed a subset of the ML questions (684 questions from 50 users). Although significant effort was involved in labelling and qualitative analysis, this subset may not accurately represent the set of all ML questions and users on Stack Overflow. Third, manual labeling is subjected to the inherent biases of human judgement. To reduce this threat, we employed two annotators and extensively discussed any disagreements. Finally, for most of the labelling tasks, we created the labels (or categories) ourselves since no previous categorization was available. To reduce any unconscious biases that might influence the categorization, we created the labels before performing any analyses and did not change the set of labels during or after analyses.

VI. RELATED WORKS

In this section, we summarize existing studies (albeit a few) on understanding developer challenges with ML, studies

exploring unanswered questions on Stack Overflow, and other software engineering related Stack Overflow studies.

A. Challenges in Developing ML Applications

As Gillies et al. [25] recognize ML is often conceived in an "impersonal" way, where ML applications are designed as standalone units trained from passively collected data. They call for a human-centered perspective, where ML workflows are situated within human working practices. We argue that application developers play a significant role in realizing human-centered ML workflows. Yet, there is little focus on systematically understanding how developers learn ML techniques, incorporate those into application development, and cope with the challenges they face in the process.

Patel et al. [3], [4] study difficulties encountered by software developers in the adoption of ML techniques. By interviewing ML researchers and users, they identify three key difficulties related to understating different ML phases, understanding the relationship between data and model, and evaluating the model's performance in the specific domain. Yang et al. [26] conduct interviews to determine the challenges non-experts (not formally trained in ML) face in developing ML applications, finding that non-experts struggle to understand the ML algorithm's internal mechanism, and are baffled when modeling performance stalls, and when there is an unexpected error. These works, though valuable, do not provide insights specific to the challenges software *developers* face since the majority of the subjects in these studies were not software developers (Patel et al.'s study involved ten graduate students as ML users, and Yang et al.'s study included four software engineers out of 14 non-experts). Further, these studies were primarily interview-based and involved a few subjects.

In contrast to the studies above, we take a fundamentally different approach toward understanding the challenges of ML application development. Our approach involves quantitative and qualitative analyses of ML-related posts on Stack Overflow, a popular Q&A forum for programmers. By doing so, we study challenges specific to software developers and offer several novel findings. To the best of our knowledge, our study is the first to exploit Stack Overflow for understanding the challenges ML application developers face.

B. Unanswered Questions on Stack Overflow

Despite the developers' active engagement on Stack Overflow [27], there is an increasing number of unanswered questions over the years [19]. Further, evidence suggests that although the number of users involved in answering questions is increasing, the number of unanswered question is increasing even more rapidly in recent years [28].

Various attempts have been made to understand the factors responsible for the unanswered questions. Yang et al. [29] study the relationship between a user's reputation and the likelihood of the question being answered. They find that questions from new users are less likely to be answered compared to the experienced users. Asaduzzaman et al. [19] perform a qualitative study, finding that too short questions are most likely to be unanswered because those questions may miss the important information. Further, they show that the greatest proportion of the unanswered questions fail to attract an expert from the community. Yang et al. [30] identify that not finding a relevant expert is the biggest factor for the unanswered questions. Baltadzhieva and Chrupaa [31] study the relationship between question quality and the likelihood of being answered, finding that high-quality questions are more likely to be answered. This is because the high-quality questions are expected to draw great user attention and will make users feel more compelled to answer the question [32].

Similar to the studies above, we also argue that there is a strong connections between lack of ML experts and unanswered ML questions on Stack Overflow. In contrast to these generic studies, which cover all sorts of questions, we focus on ML related questions. By doing so, we uncover valuable findings on the type of ML questions that are difficult to answer, the distribution of questions over ML phases, and the type of knowledge required to answer ML-related questions.

VII. Conclusions

We conducted a comprehensive analysis on ML-related questions posted by software developers on Stack Overflow. We identified five key research questions, aiming to gain a deeper understanding on the novel challenges that developers may face when performing ML centric software development. Although answering these questions is of immense practical value, there is little prior work on it. Via quantitative and qualitative analyses, we reached important conclusions about ML application development. Our first key finding suggested that ML questions are more challenging to answer than other Stack Overflow questions. Second, we analyzed the challenges developers face in different phases of ML and discussed potential reasons for those. Third, we explored the most popular and challenging topics of ML, and found that most of the challenging question pertain to data preprocessing, environment setup, and model deployment. We also found an increase in the popularity and difficulty of topics related to neural networks and deep learning techniques, especially for NLP and object detection tasks. Importantly, we also found that there is a lack of ML experts on Stack Overflow. Finally, we explored the type of knowledge required to answer developers' questions, finding that implementation knowledge is essential to answer a large portion of questions, and that there may be a shortage of experts having both conceptual and implementation knowledge.

Acknowledgment

This research was supported in part by an NSF IIS award IIS-1814450 and an ONR award N00014-18-1-2875. The views and conclusions in the paper are those of the authors and should not be interpreted as representing any funding agency.

REFERENCES

[1] New Vantage Partners (NVP), "Big data and AI executive survey 2019: How big data and AI are accelerating business transformation," http://newvantage.com/wp-content/uploads/2018/12/Big-Data-Executive-Survey-2019-Findings-122718.pdf, 2019.

[2] P. Probst, B. Bischl, and A.-L. Boulesteix, "Tunability: Importance of hyperparameters of machine learning algorithms," *arXiv preprint arXiv:1802.09596*, 2018.

[3] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison, "Investigating statistical machine learning as a tool for software development," in *Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI*, 2008, pp. 667–676.

[4] K. Patel, J. Fogarty, J. A. Landay, and B. L. Harrison, "Examining difficulties software developers encounter in the adoption of statistical machine learning." in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI*, ser. AAAI, 2008, pp. 1563–1566.

[5] A. Zagalsky, C. G. Teshima, D. M. Germán, M. D. Storey, and G. Poo-Caamaño, "How the R community creates and curates knowledge: a comparative study of stack overflow and mailing lists," in *Proceedings of the 13th International Conference on Mining Software Repositories, MSR*. ACM, 2016, pp. 441–451.

[6] M. Allamanis and C. A. Sutton, "Why, when, and what: analyzing stack overflow questions by topic, type, and code," in *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR*. IEEE Computer Society, 2013, pp. 53–56.

[7] S. Baltes, L. Dumani, C. Treude, and S. Diehl, "Sotorrent: reconstructing and analyzing the evolution of stack overflow posts," in *Proceedings of the 15th International Conference on Mining Software Repositories, MSR*. ACM, 2018, pp. 319–330.

[8] A. Bacchelli, L. Ponzanelli, and M. Lanza, "Harnessing stack overflow for the IDE," in *Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering, RSSE*. IEEE, 2012, pp. 26–30.

[9] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *International Conference on Social Computing, SocialCom*. IEEE Computer Society, 2013, pp. 188–195.

[10] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving low quality stack overflow post detection," in *30th IEEE International Conference on Software Maintenance and Evolution ICSME*. IEEE Computer Society, 2014, pp. 541–544.

[11] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[12] N. Novielli, F. Calefato, and F. Lanubile, "Towards discovering the role of emotions in stack overflow," in *Proceedings of the 6th International Workshop on Social Software Engineering, SSE*. ACM, 2014, pp. 33–36.

[13] C. Treude and M. P. Robillard, "Augmenting API documentation with insights from stack overflow," in *Proceedings of the 38th International Conference on Software Engineering, ICSE*. ACM, 2016, pp. 392–403.

[14] S. Nadi, S. Krüger, M. Mezini, and E. Bodden, "Jumping through hoops: why do java developers struggle with cryptography apis?" in *Proceedings of the 38th International Conference on Software Engineering, ICSE*. ACM, 2016, pp. 935–946.

[15] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about?: a large-scale study using stack overflow," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM*. ACM, 2018, pp. 30:1–30:10.

[16] C. Rosen and E. Shihab, "What are mobile developers asking about? A large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.

[17] C. Treude, O. Barzilay, and M. D. Storey, "How do programmers ask and answer questions on the web?" in *Proceedings of the 33rd International Conference on Software Engineering,ICSE*. ACM, 2011, pp. 804–807.

[18] X. Yang, D. Lo, X. Xia, Z. Wan, and J. Sun, "What security questions do developers ask? A large-scale study of stack overflow posts," *J. Comput. Sci. Technol.*, vol. 31, no. 5, pp. 910–924, 2016.

[19] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR*. IEEE Computer Society, 2013, pp. 97–100.

[20] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: Structure and algorithms," in *Proceedings of the 16th International Conference on World Wide Web, WWW*. ACM, 2007, pp. 221–230.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[22] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining questions asked by web developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories, MSR*. ACM, 2014, pp. 112–121.

[23] J. Chang, J. L. Boyd-Graber, S. Gerrish, C. Wang, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Proceedings of 23rd Annual Conference on Neural Information Processing Systems 2009. NIPS*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 288–296.

[24] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM*. ACM, 2015, pp. 399–408.

[25] M. Gillies, R. Fiebrink, A. Tanaka, J. Garcia, F. Bevilacqua, A. Heloir, F. Nunnari, W. Mackay, S. Amershi, B. Lee, N. d'Alessandro, J. Tilmanne, T. Kulesza, and B. Caramiaux, "Human-centred machine learning," in *Proceedings of the 2016 Conference on Human Factors in Computing Systems CHI*. ACM, 2016, pp. 3558–3565.

[26] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, "Grounding interactive machine learning tool design in how non-experts actually build models," in *Proceedings of the 2018 on Designing Interactive Systems Conference 2018, DIS*. ACM, 2018, pp. 573–584.

[27] S. Wang, D. Lo, and L. Jiang, "An empirical study on developer interactions in stackoverflow," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC*, 2013, pp. 1019–1024.

[28] B. Shao and J. Yan, "Recommending answerers for stack overflow with lda model," in *Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing, ChineseCSCW*. ACM, 2017, pp. 80–86.

[29] L. Yang, S. Bao, Q. Lin, X. Wu, D. Han, Z. Su, and Y. Yu, "Analyzing and predicting not-answered questions in community-based question answering services," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 2011, pp. 1273–1278.

[30] B. Yang and S. Manandhar, "Exploring user expertise and descriptive ability in community question answering," in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM*. IEEE Computer Society, 2014, pp. 320–327.

[31] A. Baltadzhieva and G. Chrupała, "Predicting the quality of questions on stackoverflow," in *Proceedings of the international conference recent advances in natural language processing RANLP*, 2015, pp. 32–40.

[32] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, "Analyzing and predicting question quality in community question answering services," in *Proceedings of the 21st World Wide Web Conference, WWW*. ACM, 2012, pp. 775–782.