What do developers know about machine learning: a study of ML discussions on StackOverflow

Abdul Ali Bangash, Hareem Sahar, Shaiful Chowdhury, Alexander William Wong, Abram Hindle, Karim Ali
Department of Computing Science
University of Alberta, Edmonton, AB, Canada
Email: bangash@ualberta.ca, hareeme@ualberta.ca , shaiful@ualberta.ca,
alex.wong@ualberta.ca, abram.hindle@ualberta.ca, karim.ali@ualberta.ca

***Abstract*—Machine learning, a branch of Artificial Intelligence, is now popular in software engineering community and is successfully used for problems like bug prediction, and software development effort estimation. Developers' understanding of machine learning, however, is not clear, and we require investigation to understand what educators should focus on, and how different online programming discussion communities can be more helpful. We conduct a study on Stack Overflow (SO) machine learning related posts using the SOTorrent dataset. We found that some machine learning topics are significantly more discussed than others, and others need more attention. We also found that topic generation with Latent Dirichlet Allocation (LDA) can suggest more appropriate tags that can make a machine learning post more visible and thus can help in receiving immediate feedback from sites like SO.**

***Keywords*—** stackoverflow, machine learning, topic modeling

I. Introduction

The interest of software developers in machine learning has grown in recent years. This is evident, as we show later, from an increasing number of machine learning posts on stackOverflow(SO). This trend suggests that software developers are frequently employing machine learning for solving different problems, which is also supported by previous research [1]. In order to understand these trends and discussions among developers, we conduct a study on the 28,010 available machine learning posts on SO starting from 2008 to 2018. Our concern is that software research community, educators, and online programming sites need direction on what areas of machine learning need more attention for improving developers' understanding, and consequently their productivity. In the same vein, we wanted to investigate the potential of topic modeling approach for making machine learning questions more searchable through the use of appropriate tagging system on sites like SO. For the study, we utilize the SOTorrent dataset [2] and Latent Dirichlet Allocation (LDA) model. Prior work has used LDA to categorize topics on software security [3] and SO dataset has also been in various studies such as [4].

In this paper, we investigate the following research questions that help better understanding of developers knowledge on machine learning.

- RQ1: What machine learning topics are discussed on SO?
- RQ2: What exactly do the developers discuss about those machine learning topics?
- RQ3: What are the characteristics of machine learning posts considering their popularity and difficulty?
- RQ4: Do the developers tag machine learning posts correctly, and can we improve such tagging system with topic modeling?

II. Methodology

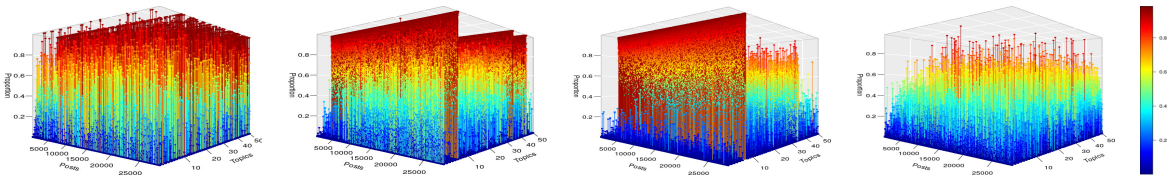
This sections describes our data collection approach, and the LDA model.

Data Collection: In this study, we use the SOTorrent dataset [2] which contains millions of SO posts from year 2008 to 2018. We used the Java SAX parser to parse the dataset and imported it into SQLite database for querying. We queried the *Posts* table in SOTorrent dataset for the `machine_learning` tag and extracted 28010 machine learning posts. The entire analysis in this paper is based on these sampled posts from the SO Torrent dataset [2].

Topic Modeling using LDA: Topic Modeling is a method of identifying topics from corpus of documents, and LDA is a prevalent method of topic modeling [5]. We employ a java-based package for topic modeling named Mallet [6]. It requires a set of documents and parameters α, β, and number of topics (K) to begin working. In addition to this, Mallet has a hyper-parameter optimization option that allows auto-tuning parameters and identify topics which better fit the data. We input 28,010 posts and run Mallet with different combinations of input parameters. Each run of the experiment return a topic-word and a document-topic matrix. The topic-word matrix is a listing of top words forming a topic whereas the document-topic matrix describes the topic weights for each input document. It is important to note that topics generated by LDA do not have a label but are rather probabilistic distribution of words only.

Topic Categorization: We ran LDA on 28,010 machine learning posts to identify their topics. We tried with two different number of topics (K=20 and K=50) for our experimental runs. For each value of K we ran the experiment by setting α and β to 0.01, 0.25 & 1.00. Figure 1 suggests that for such settings, topic weights are distributed in a way that its hard to identify leading topics. This means that some of the important topics remain hidden in the corpus, and many topics get associated with documents even when they are not related. We then ran LDA with Mallet's hyper-parameter optimization

2574-3864/19/$31.00 ©2019 IEEE
DOI 10.1109/MSR.2019.00052



(a) α/β=0.01 (b) α/β=0.25 (c) α/β=1.00 (d) α/β=auto-tune for optimization-interval=20

Fig. 1. Four runs of LDA for 50 topics with varying configurations of hyper-parameters

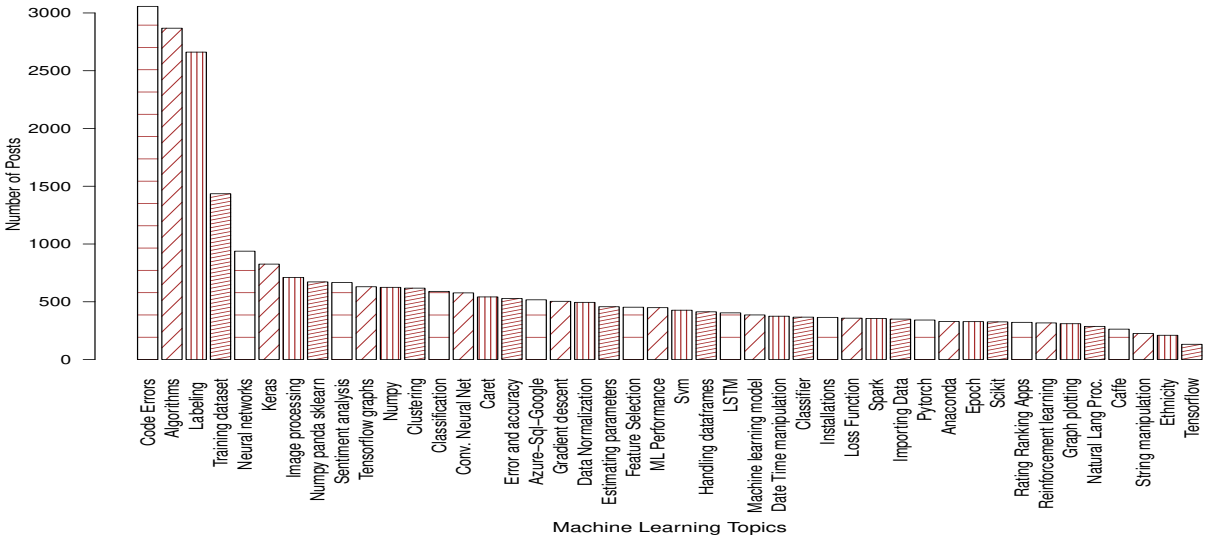


Fig. 2. Number of posts distributed across topics.

setting in which α and β are auto-tuned. Figure 1 shows the results of Mallet's auto-tuned parameters (which range from 0.00417 to 0.66029 when K=50 and optimization-interval=20). In this case, Mallet's optimizer returned topics that seem close to actual topics, hence identification of topic describing the document becomes easier.

We then assigned labels to the topics based on multiple authors' consensus. Assigning names to topics is important as LDA does not label them, and carrying out analysis on lists of words is difficult. One of the authors manually performed the task of labeling all the topics, and a separate author verified the results. In case of disagreement, the opinion of a third author was taken to resolve the conflict. It is important to mention here that we had to discard 6 out of 50 topics because the list of word were not sensible. Hindle et al. [7] also mentioned in their LDA study that labeling some topics might not be possible. In the end, we grouped similar topics together into four broad categories and assigned names to each category–finally leading to 44 topics under four broad categories. This was also done based on the consensus of multiple authors'.

We have made all the words-list of topics, graphs and manually explored questions available in replication dataset[1].

Manual Exploration: To validate whether the LDA suggested topics are useful, we randomly sampled 230 machine learning posts and manually read them. We read questions, answers and comments from each post and verify if LDA predicts the right topic for the posts—considering our understanding of the topics as the ground truths. We also note the metadata information of each post such as tags for further analysis. Manually reading posts not only serves the purpose of validation but also allows us to comment on the usefulness of LDA for classification of SO posts.

III. Evaluation and Results

The results of our analysis carried out on SOTorrent dataset are reported in this section.

RQ1: What machine learning topics are discussed on SO? Figure 2 shows the distribution of posts from SO across the 44 identified topics. Surprisingly, *Code Errors* topic is the most dominant. Our observation is that this is because of the new machine learning tools that developers are trying to adopt without enough understanding. It might be interesting

[1]https://doi.org/10.5281/zenodo.2597652

to conduct an analysis of these errors and determine their causes. Also, a major fraction of the discussions falls under the *labeling* (i.e., related to supervise learning) and *algorithms* topics followed by the *training datasets*. *Neural Networks* is another topic of discussion on SO with almost 1000 posts. This suggests that developers are getting interested in recent machine learning trends like deep learning. As we mentioned earlier, the topics that we identified were grouped into four broader categories namely: Framework, Implementation, Sub-domain (such as reinforcement learning) and Algorithm. The Framework category includes posts relevant to machine learning frameworks and according to our results some of the famous frameworks and APIs include *Numpy*, *Panda*, *ScikitLearn*, *Keras*, *Caret*, *Google-Cloud* and *Azure-Cloud*. The second category, Implementation, includes a huge number and variety of topics classifying almost 51% of our corpus. As opposed to this, not a lot of topics fall into Sub-domain and Algorithm category. The hot machine learning Sub-domains on SO are *Neural Networks*, *Image Processing* and *Sentiment Analysis* whereas the top Algorithms are *Classification* and *Clustering* algorithms as well as *Convolutional Neural Networks*.

RQ2: What exactly do the developers discuss about those machine learning topics? For answering this question we (two authors) manually analyzed 230 randomly sampled SO machine learning posts (at least 5 posts from each of the 44 topics). We identified the key issues normally being discussed among developers about machine learning. We discovered that most of the developers are interested in feature selection, the selection of more appropriate algorithms, or even how they should train the dataset. People have asked variety of questions—e.g., what features are best for a certain task and how outliers can be removed from their datasets. However, lack of answers to these questions show unavailability of community support for these topics. For most of the posts there is only one answer at most and some of them only had comments. Interestingly though, most of the *Classification* questions such as those on performance of classifiers are well-answered and question makers look satisfied [2]. We also note that theoretical questions on SO related to theoretical concepts of machine learning tend to remain unanswered and people are more inclined towards answering programming related questions. This is not surprising given that SO is mostly relevant for programming related discussions. We also noticed that developers mostly care about short term solutions, such as asking for an API that can solve their problems [3]. Many developers face issues in identifying the right format of their input data files which is used by machine learning models [4]. Some developers lack the basic understanding of partitioning data into training, validation, and testing and the concept of over-fitting [5]. The discussions on topics like computer vision and performance issues seem more difficult to the developers

[2]https://stackoverflow.com/questions/47968453/
[3]https://stackoverflow.com/questions/47197152/
[4]https://stackoverflow.com/questions/47062970/
[5]https://stackoverflow.com/questions/20547540/

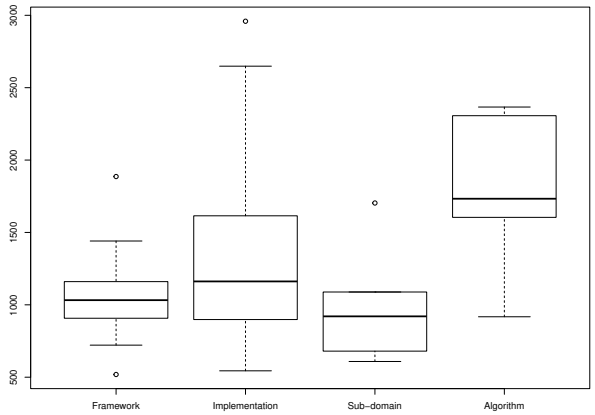


Fig. 3. Distribution of View Count(Y-axis) of Posts by Category(x-axis)

when compared to other topics [6]. These observations are useful because they show that many developers are trying to use machine learning in their software without proper understanding, which can be frustrating for the end-users. This indicates that developers need better introductory machine learning training and artificial intelligence training. The remaining examples are available in replication dataset.

RQ3: What are the characteristics of machine learning posts considering their popularity and difficulty?

Our analysis helped us discover some important traits of machine learning posts which we discuss here. We also compare machine learning posts with the rest of SO posts. Figure 4 shows that developers are asking more machine learning questions than before, and that is true for all four broad categories defined in the study. This is a direct evidence for the growing interest of developers in machine learning. However, we observe that the number of views of a question vary depending on the question category, as presented in Figure 3. One more question on this is, *are machine learning questions different than others in terms of views and accepted answers?* To answer this we compared the view counts and answer counts of machine learning posts with posts that are not related to machine learning. The non-machine learning posts were randomly sampled from SO posts (excluding machine learning) between year 2008 and 2018. From each year, we sampled non-machine learning posts equivalent to the number of machine learning posts found on SO in that year. For example in 2010 there were 301 machine learning posts on SO; so we randomly sampled 301 non-machine learning posts from that year for comparison and so on. As a result, we extracted 28,010 non-machine learning posts spanning 11 years. We then compared view counts and answer counts of machine learning and non-machine learning posts using Kolmogorov-Smirnov test and found the differences for both variables to be statistically significant at α=0.01 (p-value= 2.2e-16 for both). Further analysis on answer count reveals that machine learning questions are not answered frequently. Almost 56% of the

[6]https://stackoverflow.com/questions/52735975/

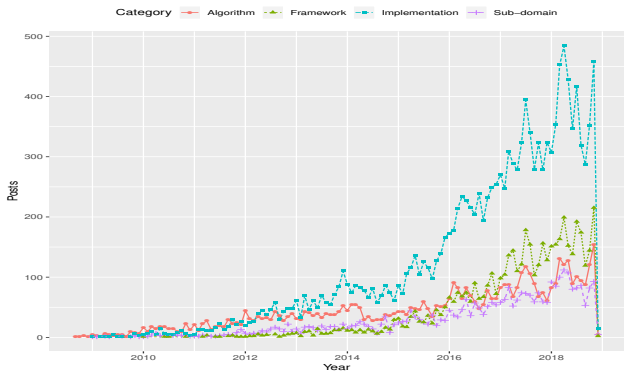


Fig. 4. Number of posts for each category over time

machine learning questions on SO do not have an accepted answer whereas 19% have no answer at all. This could be attributed to the difficulty of questions or might be because machine learning questions do not reach the right people, and thus do not have enough answers.

Our results suggest that machine learning questions are relatively harder to answer, and demand more work from the research community, similar to the one we investigate next.

RQ4: Do the developers tag machine learning posts correctly, and can we improve such tagging system with topic modeling? It is important to understand if the developers written tags for machine learning posts are accurate, because this is directly related to the number of views and answers a post can receive [8]. We also evaluate how accurate the LDA model is to asses its potential in recommending useful machine learning tags. For this, we randomly sampled (as mentioned in Section II) and read 230 posts from SO related to machine learning and started writing the tags based on two of the authors' consensus (without looking what tags are there on SO and what tag was suggested by LDA for a given post). If our written tag for a post matches with LDA, then we consider LDA to be correct for that post. We identify that out of 230 posts that we read, 151 were correctly labeled by LDA giving us an accuracy of 66.5%. The accuracy increases to 72% if we remove some outlier topics such as tensorflow and ethnicity. To our surprise, we found that 85 of the posts that were correctly labeled by LDA do not have the tags suggested by LDA. This indicates that many SO users do not have the domain knowledge for writing appropriate machine learning tags. This can be one of the explanations for the lack of community support we observed in RQ3 for machine learning posts. As an example a user asked *"how can I retrieve an attribute which influence the result in a dataset?"*. Here they provided tags like python, prediction but never added the tag of feature selection. As a result, their post might go unnoticed by feature selection experts. So we manually added the tag of *feature selection*, the topic which was suggested by LDA for this very post[7]. The LDA suggested tag for that post got immediately accepted by the SO community. While writing on this paper one of the authors made a new account at SO and manually applied tag edits suggested by LDA. One is allowed to apply five edits at most before someone can review it from the community. Out of our 28 edits that we made so far, 16 were accepted while 9 were rejected (others are not reviewed yet). The main reason of rejection was not necessarily because the tags were wrong, but for *"Please explain why these edits are necessary, rather than just doing them to get an "easy" 2 rep"*. Even the rejected tags got acceptance from at least one reviewer out of three. However, this acceptance of tags that were suggested by LDA is encouraging, and implies that online community can potentially adopt such methodologies for tagging machine learning related discussions more appropriately.

[7]https://stackoverflow.com/questions/47404605/

IV. Threats to Validity

Internal Validity Our study conducts analysis of 28,010 posts from SO. However there might be posts on SO that are about machine learning but do not have that tag. We have not taken those posts into account.

Conclusion Validity The manual labeling of topics is problematic, but we minimized this threat to some extent by taking votes from 2 of the authors. Furthermore we reported values of all LDA parameters in our paper to allow replication.

V. Related Work

Pinto *et al.* [9] analyzed SO posts to understand what developers know about software energy consumption. Similarly, Yang *et al.* [3] investigated security related questions from SO. These types of studies are helpful for educators and researchers to understand where should they focus to help developers on a certain topic. The closest to our work is the study by Patel *et al.* [10]. However, that study focused only on statistical machine learning with expert researchers and selected number of developers.

VI. Conclusion

In this paper, we studied the developer discussions on machine learning on the famous StackOverflow site. After analyzing 28,010 machine learning posts, we employed topic modeling technique to identify key areas that are of interest to developers. Our analysis based on LDA revealed 44 topics (classified into four categories) including Algorithms, Classification, and Training datasets categories that are frequently discussed by the developers. Our results also indicate that in spite of the growing interest, developers lack proper introductory understanding of machine learning, and unfortunately they do not receive enough feedback from community. These results are directly helpful for educators and researchers alike as it is evident that more introductory education in machine learning should be given to developers is required. We also showed the potential of topic modeling approach towards a better tagging system for machine learning discussions. Such a tagging system would help developers to reach the right people in the community, and would possibly bring earlier feedback on their questions.

REFERENCES

[1] A. McIntosh, S. Hassan, and A. Hindle, "What can android mobile app developers do about the energy consumption of machine learning?" *Empirical Software Engineering*, Jun 2018.

[2] S. Baltes, C. Treude, and S. Diehl, "Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets," *arXiv preprint arXiv:1809.02814*, 2018.

[3] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun, "What security questions do developers ask? a large-scale study of stack overflow posts," *Journal of Computer Science and Technology*, vol. 31, no. 5, pp. 910–924, 2016.

[4] S. A. Chowdhury and A. Hindle, "Mining stackoverflow to filter out off-topic irc discussion," in *Proceedings of the 12th Working Conference on Mining Software Repositories*. IEEE Press, 2015, pp. 422–425.

[5] J. C. Campbell, A. Hindle, and E. Stroulia, "Latent dirichlet allocation: extracting topics from software engineering data," in *The art and science of analyzing software data*. Elsevier, 2015, pp. 139–159.

[6] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002.

[7] A. Hindle, C. Bird, T. Zimmermann, and N. Nagappan, "Relating requirements to implementation via topic analysis: Do topics extracted from requirements make sense to managers and developers?" in *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*. IEEE, 2012, pp. 243–252.

[8] A. K. Saha, R. K. Saha, and K. A. Schneider, "A discriminative model approach for suggesting tags automatically for stack overflow questions," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, 2013, pp. 73–76.

[9] G. Pinto, F. Castor, and Y. D. Liu, "Mining questions about software energy consumption," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014, 2014, pp. 22–31.

[10] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison, "Examining difficulties software developers encounter in the adoption of statistical machine learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, 2008, pp. 1563–1566.