Analyzing Stack Overflow Posts using LDA method

Ankit Kumar
Sridhar Chimalakonda
Akhila Sri Manasa Venigalla
cs20m004@iittp.ac.in
ch@iittp.ac.in
cs19d504@iittp.ac.in

ABSTRACT

Software developers frequently solve development issues with the help of question and answer web forums, such as Stack Overflow (SO). Stack Overflow (SO) is one of the most frequently used websites. While tags exist to support question searching and browsing, they are more related to technological aspects than to the question purposes. These websites serve as knowledge sharing platforms where Subject Matter Experts (SMEs) and developers answer questions posted by other users. It is effort intensive for developers to navigate to right posts because of the large volume of posts on the platform, despite the presence of existing tags, that are based on technologies. Both researchers and developers can benefit from our approach and results to classify posts into the three question categories. For instance, our approach could help to improve existing code recommender systems using Stack Overflow (SO), such as Sea hawk. It could allow recommenders to filter the posts according to the three question categories, and thereby improve the accuracy of their recommendations.

Furthermore, our approach can improve existing research on analyzing and identifying topics discussed on SO posts. With our question categories, an orthogonal view on the topics discussed on SO is provided. This enables researchers to investigate the relationships between topics and reasons and thereby study the what and why of discussions on SO. in this paper we found out three topic question categories which is-

- *Topic*1 - Web development
- *Topic*2 - Cloud containers
- *Topic*3 - Cloud computing services

KEYWORDS

Stack Overflow (SO), Latent Dirichlet Allocation (LDA), natural language processing

1 INTRODUCTION

Stack Overflow (SO) is one of the most frequently used websites with about 15M visits every day. With a user base of 10M users, about 7.3K questions are posted per day. It comprises of about 18 million questions, of which 71 percent are answered. These questions correspond to various technical categories, tools, libraries and are tagged into at most 5 of 54K tags present on the website. these tags don't classify questions based on the context in which they are asked. The context would capture situations pertaining to conceptual understanding, issue resolving and so on. Recent studies have aimed at classifying questions on stackoverflow (SO) based on their context and arrived at almost similar taxonomies of categories. it automatic categorization by topic modeling using LDA. and also we manually categorize. by question category. For example, developers can use our models to find web development specific challenges by question category. Also, the classification can be leveraged by researchers to build better SO-based recommenders. Our results have several implications for developers and researchers. Integrating our models into SO, developers can search In summary, the main contributions of this paper are - A taxonomy of 3 question categories. A manually labeled data set that maps 100 questions to 3 question categories. An approach to automatically classify posts into the 3 question categories and we use Application of NLP techniques - Latent Dirichlet Allocation (LDA) and Machine learning (ML) classifier Support Vector Classifier (SVC) to classify SO posts.

Behind the motivation is I read paper that title was -Learning with comments: An analysis of comments and community on Stack Overflow they categories the comments of stackoverflow post so I thought that I will experiment on stackoverflow question and I will find the topic which is depend on LDA techniques. I also read the paper title was - SOTagger - Towards Classifying Stack Overflow Posts through Contextual Tagging which was done by my mentor, in this paper they use LDA technique and they come up SOTagger application.

The contributions of this work are as follows:

- We investigate the distribution of questioners and answerers and investigate their behaviors.
- We employ topic modeling to assign topics to tens of thousands of questions from StackOverflow.
- we calculate Performance of our models that is classifing the stackoverflow SO posts into the 3 question categories?

The structure of this paper is as follows. In Section 2, we describe Dataset information on StackOverflow and topic modeling (Latent Dirichlet Allocation). In Section 3, we describe Experiments in the Empirical Study. in Section 4, we describe Results. in Section 5. we describe Limitation of the study. in Section 6. we describe Related Work in section 7. Finally, we describe conclusion with future work.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATASET

Extract DataSet- I extract the first 500 Stack Overflow (SO) post because of with my IP address only allow 500 post per day. So I download each day 500 Stack Overflow (SO) post until I got 10k Stack Overflow (SO) post.For every call, the API would return 500 randomly elected Stack Overflow (SO) post or we can choose by date.We use these questions and their answers in our study. I scrape the some specific data so I fetch five data item from Stack Overflow (SO) posts.

- question Title
- question(full)
- tags
- Question(UserId)
- Answer(UserId)

after scraping data I combine the all csv file using *mergecsv_code.py* using this program it will combine into one csv file which name is *merge_data_stackoverflow.csv* file. I use StackAPI library for the data scraping. I build code using python language using of this I use the StackAPI library and scraping the data.

To perform categorization of SO posts we downloaded datascrapper.csv file which get by the file datascrapper.py. We considered a subset of this file that constituted 10K Stack Overflow posts under Body column that resulted in a dataset of 10K posts. and filtered out questions based on question column.

Data Pre-processing: Data present in question column in csv file was considered for preprocessing. We will perform the following steps:

- Tokenization: Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation.
- All stopwords are removed.
- We used Gensim here, use (deacc=True) to remove the punctuations.
- Stemming(lemmatization using spaCy) Words are stemmed words are reduced to their root form. The advantage of this is, we get to reduce the total number of unique words in the dictionary. As a result, the number of columns in the document-word matrix will be denser with lesser columns. From this we can expect better topics to be generated in the end.

Topic modeling

Step 1- Latent Dirichlet Allocation Model- We applied LDA to perform topic modeling. The LDA topic model algorithm requires a document word matrix as the main input. So I created this using using CountVectorizer. I have configured the CountVectorizer to consider words that has occurred at least 10 times, remove built-in english stopwords, convert all words to lowercase, and a word can contain numbers and alphabets of at least length 3 in order to be qualified as a word.Everything is ready to build a Latent Dirichlet Allocation (LDA) model. Let's initialise one and call $fit\_transform()$ to build the LDA model. We use here online variational Bayes algorithm for findout the best parametres. LDA model that categorizes given data into 4 topics.

Step 2- Naming Topics. We identified contextually useful keywords in each of the 3 topics, and used them to identify and name topics. Step 5 - Append Labels to Dataset. The LDA model provided us with a topic-document correlation matrix, where document refers to content of one post. This matrix contained probabilities of every identified topic for each document. We then classified posts in the dataset into topics based on the dominant topic from correlation matrix which had the highest probability. Step 6 - Prepare a Machine Learning model.

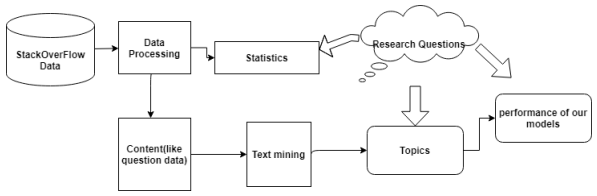

Figure 1: Empirical Study Framework

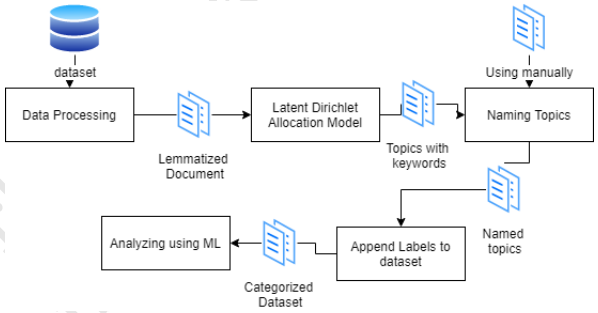

Figure 2: Overview of this research

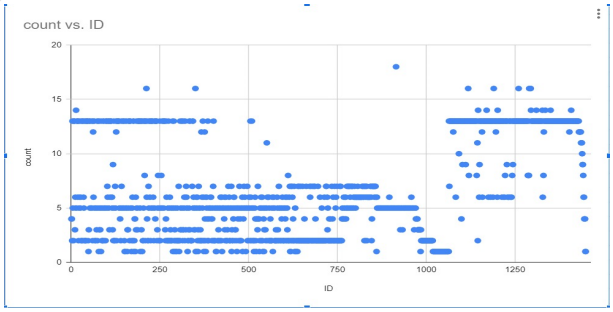

Figure 3: graph1

3 EXPERIMENTS IN THE EMPIRICAL STUDY

Research Question (RQ1)-analyzing the distribution of questions related to user in stack-overflow. We plot the graph of questioners in Figure 3. The graph shows the number of users that ask a given number of questions, and its y-axis is count. From the graph, we notice that most users only ask one question. Only some of the uses ask two or more questions. The number of users that ask questions reduces exponentially as we consider a higher number of posted questions. Only very less users ask more than 13 questions.

Experiment-

I took the data from *merge_data_stackoverflow.csv* file and from

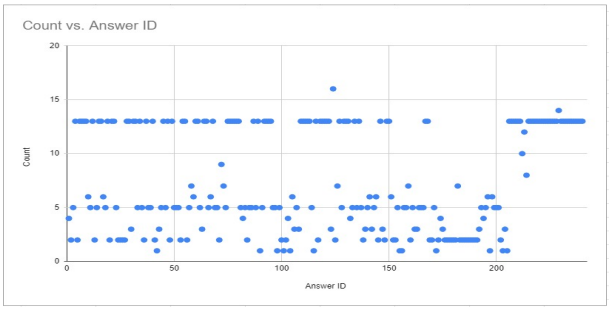

Figure 4: graph2

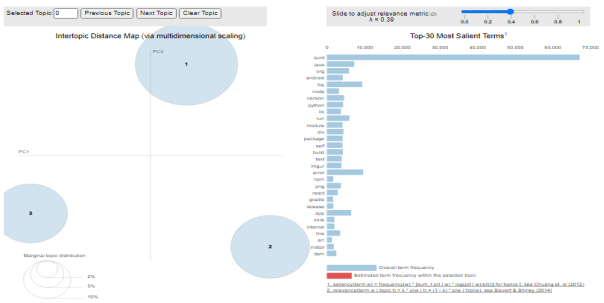

Figure 5: graph3

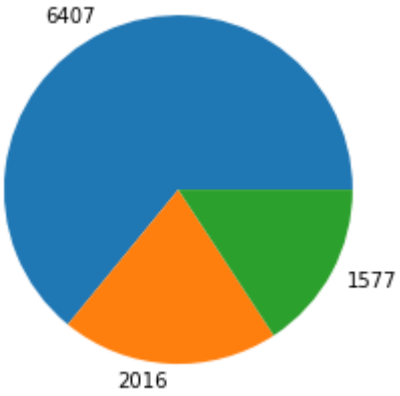

Figure 6: graph4

4th column(*question_user_id*) we take that column all values and draw the graph using Microsoft Excel.

Research Question (RQ2)-analyzing the distribution of answers related to user in stack-overflow. We plot the graph of answers in Figure 4. The graph shows the number of users that answer a given number of questions and its y-axis is count. From the graph, we notice that most users only answer one question. Only about some of the users answer two or more questions. Very less users give answer more than 10 questions.

Experiment-

I took the data from *merge_data_stackoverflow.csv* file and from 5th column(*answer_user_id*) we take that column all values and draw the graph using Microsoft Excel.

Topic	Words	Question
Web development	Javascript, html, css etc	6407
cloud containers	Docker, contatiner etc	2016
cloud computing services	Amazon Web Services, azure etc	1577

Figure 7: Table1-question categories

Research Question (RQ3)- What is the performance of our models to classify SO posts into the 3 question categories? A model with higher log-likelihood and lower perplexity is considered to be good. using this we can make better version of the LDA models and we can improve the model using different function, here we are going to find out the performance of this model.

Research Question (RQ4) What are the topics of question categories of stackoverflow (SO) posts? here I run the our LDA model on given data and it will give us 4 topics using of then we manually read the keywords and decide what topic name it should be and we build the model like if we give the question data in text variable it will give us the probability of all four topics which will get the higher probability then we take the high probability topic and assign that topic to the particular question. and I create also which is automatically assign the topic to each question.

4 RESULTS

Explain the results of each research questions.

Research Question (RQ1) result-

analysing the distribution of questions related to user in stack-overflow. The result shows that there are few regular questioners on Stack Overflow. This is possibly because many questions have already been asked before and users could and answers to them by just looking into the various pages on Stack Overflow or other question and answer sites via search engines.

Research Question (RQ2) result-

Compared with the distribution of questioners, the distribution of answers is different. The number of users that answer a substantial number of questions(> 10) is more than the number of developers that ask a substantial number of questions (> 10). This may imply that many users on StackOverflow are interested in contributing to the community and are not solely interested in getting his or her questions answered.

Research Question (RQ3) result- The results show that our models can classify SO posts into the 3 question categories with performance of 1.Log Likelihood Score: -1584582.5417850607, 2.Model Perplexity: 743.7041895978797

Research Question (RQ4) result-

To answer this research question, we run a topic modeling technique LDA on the text and code contents of the questions that users ask. We set the number of topics to be 3, and after LDA completes running, it outputs 3 topics: each topic is a set of words sorted in terms of their likelihood of belonging to the topic. LDA does not generate a meaningful label for each topic; We manually study the

words in each topic and related questions, and assign a label to the topic. Based on the above analysis, figure 7 shows the three topics with our manually assigned labels, some representative words in each topic, and the identifier of an example question that has a high probability of belonging to the topic. We assign only one topic with the highest probability to a question based on the topic probabilities assigned by LDA to the question and count how many questions belong to each topic shown in the figure 6 (graph 4). and in the figure 5 (graph3) in which given in *lda_code* there you can click on each circle it will tell us high occurrence words.

5 LIMITATIONS OF THE STUDY

Explain the limitations about the study talking about technical challenge related to experiments, data set creation. You should also mention any assumption you that you consider while conducting experiments.

Limitation is that I apply on my model only 10k stackoverflow posts instead of I have to take more large dataset. And I have to apply more better model in comparison of LDA techniques(unsupervised learning), we could also try supervised learning may it will give us high accuracy. Limitation of this model

- Uncorrelated topics (Dirichlet topic distribution cannot capture correlations)
- Non-hierarchical (in data-limited regimes hierarchical models allow sharing of data)
- Static (no evolution of topics over time)
- Bag of words (assumes words are exchangeable, sentence structure is not modeled)
- Unsupervised (sometimes weak supervision is desirable, e.g. in sentiment analysis)

I was facing lots of difficulties like scrapping the data it is very difficult to scraping the more than 500 posts so instead of this I scrape the daily 500 posts till it reach to 10k posts. I read documentation of stackAPI library and I manually scrape the data and labeling them. and for merging the all posts in the one csv file so I build the python code with this help I could merge the all csv file into one csv file. for my model coding part there is two library which they provide to build LDA model one is gensim and other one is scikit-learn I try both of them then I choose scikit-learn library for my LDA model. And installing spacy also difficult I try so many command installing the spacy, while data cleaning I also read the text that what I have to take/leave. From this I learn NLP techniques and ML techniques. So many error I face while building the LDA model.

6 RELATED WORK

The posts on SO were often used to investigate the categories and topics of questions asked by software developers in Treude et al. Were the first ones investigating the question categories of posts of SO. In 385 manually analyzed posts, they found 10 question categories: How-to, Discrepancy, Environment, Error, Decision Help, Conceptual, Review, Non-Functional, Novice, and Noise. Similarly, Rosen et al. manually categorized 384 posts of SO for the mobile operating systems Android, Apple, and Windows each into three main question categories: How, What, and Why. Beyer et al.applied card sorting to 450 Android related posts of SO and found 8 main question types: How to...?, What is the Problem...?, Error...?, Is it possible...?, Why...?, Better Solution...?, Version...?, and Device...? Based on the manually labeled dataset, they used Apache Lucene's k-NN algorithm to automate the classification and achieved a precision of 41.33percent. Similarly, Zou et al. used Lucene to rank and classify posts into question categories by analyzing the style of the posts' answers.

There are a number of software engineering studies that employ topic modeling. Asuncion et al. use LDA for software traceability. Wang et al. investigate the effectiveness of many topic modeling approaches for concern location. Chen et al. use topic modeling to and defect prone topics from Mozilla Firefox, Eclipse, and Mylyn. one of the paper they have manually created a curated data set of 500 SO posts, classified into the seven categories. Using this data set, we apply machine learning algorithms(Random Forest and Support Vector Machines) to build a classification model for SO questions. We then experiment with 82 different configurations regarding the preprocessing of the text and representation of the input data. The results of the best performing models show that our models can classify posts into then correct question category with an average precision and recall of 0.88 and 0.87 when using Random Forest and the phrases indicating a question category as input data for the training. The obtained model can be used to aid developers in browsing SO discussions or researchers in building recommenders based on SO.

7 CONCLUSION AND FUTURE WORK

In this research, we analyzing on StackOverflow posts and categories each question into one of the three topics. recently analyze 200 questions manually and label them into 3 categories. I also investigate the usage of tags in StackOverflow. We find that most developers only answer or ask one question. many users only ask questions but never answer any.Few users answer and ask many questions. There are a very less user who ask answer more than 3 questions. there are some user who ask and answer a similar number of questions. We analyzing that questions posted by users could be grouped into 3 categories based on the topic modeling technique that we use: user interface, stack trace, large code snippets, web documents, and miscellaneous.

for future work- for future work, we plan to extend this study by investigating more questions from StackOverflow and from other question and answer web sites. We plan to try various numbers of topics and various topic modeling techniques. and we can also experiment on titles of questions and find the topic from it. and we can also experiment on Stack Overflow (SO) post comment and find the topic form it.

8 ARTIFACTS

GitHub link of my Empirical study.
https://github.com/ashupipalia/Analyzing-Stack-Overflow-Posts

9 REFERENCES

1.https://ksiresearch.org/seke/seke19paper/seke19paper_67.pdf
2.https://dl.acm.org/doi/abs/10.1145/3196321.3196333
3.https://www.researchgate.net/publication/262239821_An_empirical_study_on_developer_interactions_in_StackOverflow

4.https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

5.https://core.ac.uk/download/pdf/286030414.pdf