SOTagger - Towards Classifying Stack Overflow Posts through Contextual Tagging

Akhila Sri Manasa Venigalla
Indian Institute of Technology
Tirupati, India
cs18m017@iittp.ac.in

Chaitanya S. Lakkundi
Indian Institute of Technology
Tirupati, India
cs18s502@iittp.ac.in

Sridhar Chimalakonda
Indian Institute of Technology
Tirupati, India
ch@iittp.ac.in

***Abstract*—There is an ever increasing growth in the use of Q&A websites such as Stack Overflow (SO), so are the number of posts on them. 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. Tagging these posts based on their context and purpose might help developers and SMEs in easily identifying questions they wish to answer and also in identifying contextually similar posts. To support this idea, we propose *SOTagger* as a prototype plug-in for Stack Overflow to tag questions contextually. We have considered SO data provided on *SOTorrent* and automated the identification of 6 categories of questions using Latent Dirichlet Allocation. We have also manually verified relevance of these categories. Using these categories and dataset, we have built a classification model to classify a post into one of these six categories using Support Vector Machine. We have evaluated *SOTagger* by conducting a user survey with 32 developers. The preliminary results are promising with about 80% developers recommending the plugin to others.**

***Index Terms*—Stack Overflow, Contextual Tagging, LDA**

I. Introduction

Stack Overflow (SO) is one of the most frequently used websites with about 11M 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% are answered[1]. These questions correspond to various technical categories, tools, libraries and are tagged into atmost 5 of 54K tags[2] present on the website. This tagging is done based on their technical relevance with the posted content and is used to organize posts and thus help users to browse for questions and answers concerning to particular topics such as *javascript, jquery, python* and so on [1]. However, 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 SO based on their context and arrived at almost similar taxonomies of categories. They have used various techniques such as K-NN clustering [2], automatic categorization by topic modeling using LDA and MALLET [3] and manual categorizations [1], [4]. Some of these studies have aimed to contextually categorize technology-specific questions such as questions related to Android application development [2] and mobile operating systems like *Android, Apple* and *Microsoft Windows*. However, existing tools do not categorize posts on SO platform based on context. To this end, the contributions of this paper are as follows:

- *SOTagger*[3] - a prototype plug-in that classifies posts on SO into six categories: *Conceptual, Discrepancy, Implementation, Error, Learning and MWE (Minimum Working Example)*.
- Application of NLP techniques - Latent Dirichlet Allocation(LDA) and Machine learning (ML) classifier - Support Vector Classifier (SVC) to classify SO posts.
- Evaluation of *SOTagger* with 32 professional developers and manual cross-verification of 100 posts.

II. Related Work

In the recent years, several studies have been done to analyze posts on SO, which include analyzing developers' area of interest based on questions asked [5], analyzing and suggesting tags of the questions [2] [1] [6] [7], identifying difficulties faced by developers [8], identifying trending technological topics [9], and so on. Researchers have classified posts on SO based on the context by manually interviewing software developers. In a survey conducted by Latoza et al., 179 professional software developers were asked to identify hard-to-answer questions pertaining to code that they solicit wherein 371 questions were reported. They have manually categorized them into 21 categories with 94 distinct questions, of which the 5 most frequently reported categories were - *Rationale, Intent and Implementation, Debugging, Refactoring and History of code* [10].

Studies have been conducted to investigate various question categories based on the context in which they

DOI reference number: 10.18293/SEKE2019-067

[1]https://stackexchange.com/sites?view=list#traffic

[2]http://bit.ly/SONumTags

[3]https://github.com/chaitanya-lakkundi/SOTagger

were asked. Rosen et al. manually categorized 380 posts on SO into 3 question categories based on the three interrogative words- *How, What and Why*, corresponding to three mobile operating system categories - *Android, Apple* and *Microsoft Windows* [4]. Treude et al. have manually classified 385 questions on SO into 10 categories - *How to, Decision Help, Discrepancy, Environment, Error, Conceptual, Review, Non-Functional, Novice, Noise* [1]. Although methods involving manual effort are necessary to capture ground truth, we see a need to find better ways to scale this approach such that automation is possible.

Elucidating further studies, Beyer et al. have proposed 7 question categories - *API Change, API Usage, Conceptual, Discrepancy, Learning, Errors, Review* by manually classifying 500 SO Android posts and performed automatic classification using supervised machine learning algorithms with a precision of 88% [2]. Allamanis et al. found 5 major question categories using LDA and unsupervised machine learning algorithm [3].

Insofar as the development in methods of classification is concerned, the research community has progressed from significant manual studies to automating them using machine learning algorithms and NLP techniques. Contemporary tools such as EnTAGREC++ [6], TagCombine [7] have been developed to provide tag suggestions to users when they post questions on SO. These tools suggest tags based on technologies involved in the post content. The prototype plug-in we propose, *SOTagger*, tags posts on SO based on their purpose or intent rather than considering the technologies involved. Based on the existing work on classifying posts [2] [1] [4] [3], we propose a taxonomy to tag posts contextually.

III. Proposed Taxonomy

Posts can be classified using several NLP techniques such as LDA, LSA, TF-IDF. However, inline with the existing work, we followed LDA technique.

We present six question categories that we have derived from existing studies and results obtained from LDA topic modeling. As a result of LDA topic modeling configured for 6 topics, we obtained 6 topics characterized by keywords for each topic, along with the weightage of keywords in every topic. Omitting the technical terms and considering interrogatives, it has been observed that *Topic 0* comprises of *discrepancy*, *Topic 1* contains *error*, *Topic 2* contains *how-to or implementation*, *Topic 3* contains *learning*, *Topic 4* contains *conceptual* and *Topic 5* contains *MWE* keywords respectively, as shown in Table I. These results obtained by applying LDA on SO posts indicate the presence of contextual categories in SO data. Comparing these results with the existing taxonomy discussed by Beyer et al. in [2] and other taxonomies presented in [1] [4] [3], we reorganize few categories in the existing literature and arrive at labelling five of these six topics as *conceptual, discrepancy, implementation, error and learning* respectively. We observed that many of the posts on SO contained code snippets, which could indicate that users post questions containing code to reproduce the bug they are facing. Such code snippets serve as *Minimum Working Examples (MWE)*[4], which is proposed as another category *MWE*. We observe this naming to be inline with work proposed by Allamanis et al. [3]. Each post can be classified into one or more of these six categories.

TABLE I
Taxonomy of Question Categories

S.No.	Topics	Keywords
1	Conceptual	What is use/difference, Is there a way, Is it possible[2]
2	Discrepancy	doesn't work, tried to, have/facing problem, before upgrade previous version [2]
3	Implementation	How to implement [4] [3] [1]
4	Error	Exception, error [2]
5	Learning	suggest, tutorial, where can I find [2]
6	MWE	for this code, code tags

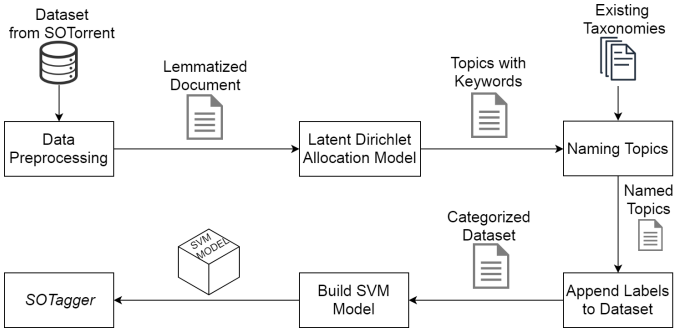

Fig. 1. Overview of Approach for *SOTagger*

IV. Design Methodology

We followed a six step approach in designing a contextual classification model as shown in Fig 1.

Step 1 - Extract DataSet. To perform categorization of SO posts, we downloaded *Posts.xml* file avilable on *SOTorrent*[5]. We considered a subset of this file that constituted 100K Stack Overflow posts under *Body* column and filtered out questions based on *PostTypeId* column that resulted in a dataset of 20K posts.

Step 2 - Data Preprocessing. Data present in *Body* column whose *PostTypeId = 1* was considered for preprocessing. We considered English stop words provided by NLTK library and omitted interrogative words from the list of stop words keeping in view, the taxonomy proposed. We processed the data for stop word, punctuation removal and lemmatization using *spaCy*.

[4]https://stackoverflow.com/help/mcve
[5]https://zenodo.org/record/2273117

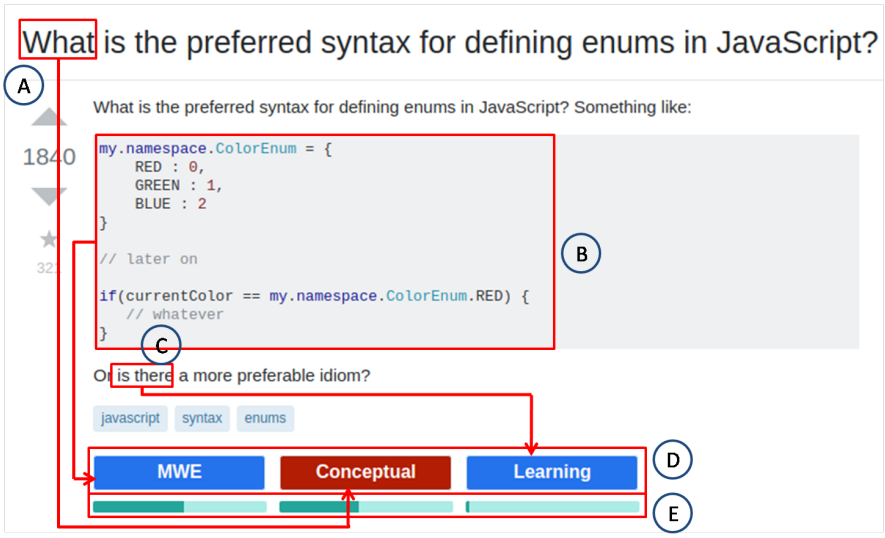

Fig. 2. A Snapshot of *SOTagger*

Step 3 - Latent Dirichlet Allocation Model. We applied LDA to perform topic modeling. We primarily created a dictionary of lemmatized words and then created a corpus of these words with their frequency of occurrence. Considering this corpus, we generated an LDA model that categorizes given data into 6 topics.

Step 4 - Naming Topics. Based on existing taxonomies in the literature [2] [1] [4] [3], we identified contextually useful keywords in each of the 6 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 - Build SVM Model. We applied various machine learning classification algorithms such as *Linear SVC, Logistic Regression, Multinomial Naive Bayes, Random Forest Classifier* to arrive at the best classification model on available dataset with 75% train and 25% test data. We observed that SVC was able to classify the given data set with higher accuracy (78.5%) than other models. Based on this, we designed SVC model and pipelined to CalibratedClassifierCV to get prediction probabilities.

V. Development of *SOTagger*

This plug-in has been developed as an extension to *Google Chrome* to support classification of posts on SO. It tags posts on SO based on their context. *SOTagger* reads SO posts on the page and extracts questions from these posts which are fed into previously developed ML classification models using SVM classification. This model outputs the categories of specific posts along with associated probabilities which are presented as tags below the posts on SO platform.

A snapshot of *SOTagger* is shown in Fig 2 for a sample post on SO. Tags corresponding to context of the question are displayed below the post as shown in *[D]* of Fig 2 and are arranged in decreasing order of probability. The probability with which a post is tagged into each of the displayed categories is represented by a bar as depicted in *[E]* of Fig 2. According to *SOTagger*, this post is classified as *MWE* category with highest probability. As pointed in *[B]* of Fig 2, presence of code segment justifies classification of the post into *MWE* category. Presence of *What* keyword as highlighted in *[A]* of Fig 2, contributes to *Conceptual* tag, with a lesser probability than *MWE* tag. *is there* phrase represented by *[C]* of Fig 2 contributes to *Learning* category, with least probability.

However, the keywords or phrases demonstrated in Fig 2, are for the purpose of analyzing the correctness of *SOTagger*, but are not the only basis for classification. Actual classification was based on *NLP* and *ML* techniques that have been used in development of *SOTagger*.

VI. Evaluation and Results

We evaluated *SOTagger* by conducting a user survey with 32 professional developers with a development experience ranging from 2 years to 19 years.

The participants were asked to use *SOTagger*, navigate to SO website and analyze the contextual tags added by *SOTagger*. A user survey was conducted with the help of five point Likert scale, containing a questionnaire as provided in Table II.

Apart from user survey, we manually evaluated[6] contextual tags of about 100 random posts on SO tagged by *SOTagger* and obtained an accuracy of 77%. The

[6]https://git.io/fjC83

results of our survey indicate, *SOTagger* had a good user-friendly interface (82% in Q1). In Q2, about 85% of participants have agreed that *SOTagger* has appropriately tagged the posts. The ratings in Q3 and Q4 indicate that *SOTagger* has helped about 80% of participants in faster browsing of posts on SO and that the experiment has been considerably interesting (81% in Q4). In Q5, most of the participants have agreed that they would recommend *SOTagger* to their peers (83%). .

TABLE II
QUESTIONS IN SURVEY USING A 5-POINT LIKERT SCALE.

Q1: How easy was it to use *SOTagger* interface?
Q2: *SOTagger* has tagged SO posts correctly based on their context.
Q3: *SOTagger* has helped me in quick browsing of posts based on context.
Q4: *SOTagger* has kept the whole experiment interesting and informative.
Q5: I will recommend *SOTagger* to my peers.

VII. Threats to Validity

We have manually examined top 20 posts based on probability values in each of the 6 topics generated by *LDA* technique to assign topic name. This could be inaccurate considering limited number of posts examined.

To understand the accuracy of classification, we randomly browsed 100 posts on SO. We realize that examination of 100 posts in total is not enough to get an overall idea about the accuracy of classification. During the creation of LDA model, we tweaked a few parameters such as chunk size and number of passes which resulted in different statistical distribution of topics. Some of the distributions were imbalanced and biased towards one particular topic. We selected those parameters which resulted in a nearly Gaussian distribution. We assume that LDA model which classifies data in Gaussian distribution performs better than other models. However, initial results show that accuracy of trained LDA model is around 70%, but with scope for experimenting with other distributions. The machine learning model has been trained on a dataset of 20K questions, however we should consider a larger number of posts from SO to improve our approach.

VIII. Conclusion and Future Work

In this paper, we presented *SOTagger*, a prototype plug-in to SO that tags questions on SO based on the purpose for which they are asked. We performed LDA topic modeling on data set available on *SOTorrent* to identify categories. We labelled the resultant LDA topics by harmonizing the existing taxonomies. We presented 6 question categories, independent of technical aspects involved in the questions. We then labelled question posts in the dataset into one or more of the 6 categories. We applied SVC on the labelled dataset to obtain machine learning classification model which was integrated into the plug-in to support tagging of posts on SO.

As a part of future work, we plan to extend *SOTagger* to display contextual tags of posts on SO landing page by training machine learning model only over titles of questions. We plan to work in the direction to improve levels of taxonomy from single level presented in the paper to multiple levels and display the same as a part of detailed contextual tagging. We could conduct an experiment to check whether we get better results by considering the opening and closing statements of SO posts.

Questions tagged with *MWE* could be of greater use for future research. Researchers interested to understand and analyze code provided by users when posing questions can easily find questions with this tag. We envision that future work based on this paper may include clustering posts classified as *MWE* to automatically find bugs, combine co-occurring tags to formulate new tags and so on. Also, several empirical studies on SO posts such as understanding code quality, misuse of code snippets and automatic bug reporting could be conducted.

References

[1] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web?: Nier track," in *2011 33rd International Conference on Software Engineering (ICSE)*. IEEE, 2011, pp. 804–807.

[2] S. Beyer, C. Macho, M. Pinzger, and M. Di Penta, "Automatically classifying posts into question categories on stack overflow," in *Proceedings of the 26th Conference on Program Comprehension*. ACM, 2018, pp. 211–221.

[3] M. Allamanis and C. Sutton, "Why, when, and what: analyzing stack overflow questions by topic, type, and code," in *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 2013, pp. 53–56.

[4] 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.

[5] R. K.-W. Lee and D. Lo, "Github and stack overflow: Analyzing developer interests across multiple social collaborative platforms," in *International Conference on Social Informatics*. Springer, 2017, pp. 245–256.

[6] S. Wang, D. Lo, B. Vasilescu, and A. Serebrenik, "Entagrec ++: An enhanced tag recommendation system for software information sites," *Empirical Software Engineering*, vol. 23, no. 2, pp. 800–832, 2018. [Online]. Available: https://doi.org/10.1007/s10664-017-9533-1

[7] X.-Y. Wang, X. Xia, and D. Lo, "Tagcombine: Recommending tags to contents in software information sites," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 1017–1035, 2015.

[8] A. Joorabchi, M. English, and A. E. Mahdi, "Text mining stackoverflow: An insight into challenges and subject-related difficulties faced by computer science learners," *Journal of Enterprise Information Management*, vol. 29, no. 2, pp. 255–275, 2016.

[9] 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.

[10] T. D. LaToza and B. A. Myers, "Hard-to-answer questions about code," in *Evaluation and Usability of Programming Languages and Tools*. ACM, 2010, p. 8.