NBSL:A Supervised Classification Model of Pull Request in Github

Song Yu and Li Xu and
Zhifang Liao* and Yanbing Li
School of Software,
Central South University, Changsha, China
Email:xulili@csu.edu.cn

Yan Zhang
School of Engineering
and Built Environment,
Glasgow Caledonian University, UK
Email:yan.zhang@gcu.ac.uk

Jinsong Wu
Department of Electrical Engineering,
Universidad de Chile, Santiago, Chile
Email:wujs@ieee.org

***Abstract*—A lot of Pull Requests (PRs) appear in Github everyday, and thus it is a very important work to review these PRs quickly in Github. Labeling PRs according to the PRs classification can improve the success rate and the review efficiency. However, recent research works have shown that most of the PRs are not labeled, and if the PR is labeled, it is done manually. To solve this problem, we propose a supervised classification model combined with supervised topics model and Naive Bayes classifier, which can make the PR be classified automatically. The method creates a one-one relationship between labels and PRs, and the approach classifies most PRs automatically with the only label which record the closest topic of PRs. The experimental results show that the proposed model can reach a precision of 60% in majority situation. The proposed model may support a better result via adjusting the parameters in case.**

I. Introduction

In recent years, Github[1] has become the largest code host in the world with more than 15 million developers collaborating across more than 38 million repositories, numerous open source projects hosted on it. In Github, developers use Pull Request(PR) mechanism to submit code revision everyday. PR mechanism [10] is presented in Figure.1.

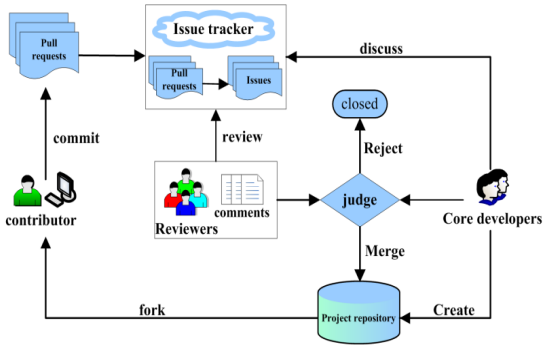

Fig. 1. the overview of pull request mechanism

When a project presents in the project repository, everyone can fork the project in his own repository. If one(we call him as a contributor) implements some new features or fixes bugs, the patches are packaged as a PR submitted to the issue tracker. The system opens a new issue for this PR,and adds the issue to an awaiting list to be reviewed. The responsible mangers of the core teams take all of the reviewers comments into consideration, then merges or rejects that PR. Thus, it is important for PR mechanism to improve the efficiency of PR review.

At present, the research works of PRs mainly focused on the submissions [7], reviews [10] and evaluations [9]. Such as finding out the factors that affect the success commit of PRs,exploring the impact of social media that used in Github. Researches also have shown the type of PR has a significant influence on the committing, review and evaluation process. The reviewers or integrators find the PR whether related to him or not with PRs labels, then improving the efficiency of PR reviews. A contributor also can check the PR he wants to commit whether has been done or not, to avoid duplicating works. However, the category labels assignment of PRs is now organized manually in GitHub, and most of PRs without labels. Therefore, we propose a supervised classification model(NBSL) combining supervised topics model with the Naive Bayes classifier to complete the classification of PRs. According to the classifications, reviewers are assigned to the corresponding PRs rapidly.

The remainder of this paper is structured as follows: Section II presents the related work, including the works on PRs and different methods used in text classification. We describe our research models in Section III. In Sections IV, we provide the experiment design, results and validation. Then at last in Section V, we present conclusions and possible future works.

II. Related work

At present, the research works on PRs have been still in the stage of development, mainly focused on the submitting, review and evaluation process. To recommend experts for code review process, Yue Yu et al.[10] used social relationship among developers to recommend code reviewers for Github systems. Yang Zhang et al.[11] used statistical analysis techniques to analyze the use of "@" in PR, to explore the impact of the "@" on the PR review process. Yue Yu et al.[9]applied the method of quantitative analysis and linear regression analysis to find out the factors that affect the processing of PR. To explore what factors will affect the success of PRs, Chanchal

*Zhifang Liao is corresponding authors for this paper.

[1]Github.https://github.com/

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

Roy et al.[7]conducted a comparative study to analyze the various features of PRs between the failure and success. And In order to explore the type of PR whether impact on their success or not, they proposed JGibbLDA topic generation model to add topic label for PRs.

A variety of approaches have been presented to classify different texts. Ying Fu et al.[4]employed a semi-supervised topic model learning algorithm to complete automatical classification of software changes. The professional vocabularies which generated in the field of software changes were used as their tag library. Then they made comparison between classified and unclassified documents by cosine similarity method.G.Antoniol et al.[1] made use of the topic model of vocabulary-based, to reach the correct classification of the error message. Natthakul Pingclasai et al.[6] used the topic model in error message classification, and compared the impact on the topics and vocabulary, and finally proved that the efficiency of topic-based classification method more significantly.

For the purpose of classification of PRs, we propose a novel and lightweight approach that combines supervised topics model with Naive Bayes to label and classify each PR in Github.

III. Research Methodology of NBSL

The LDA [2] model is an unsupervised model. It has been shown that the approach based on LDA is not a good idea to the classification [4], such as the implicit topics, the differences between nature-language texts and training messages. Considering these reasons, we use domain knowledge of issues, together with PR topic to train supervised topics model by LDA(SLDA) and then we can get the supervised topics with labels, as shown in Fig.2. According to these topics, we get the PR categories by Naive Bayes. And with the PR categories, reviewers can find the PR he would like to review more easier and more efficiency.

In this paper, we propose a Naive Bayes classification model based on SLDA method (NBSL)to achieve the goal of classification. Two phases are necessary for building the supervised PR classification model: (1)generate the SLDA model and (2)building the Naive Bayes classifier based on supervised topics. These steps are showed in detail in Fig.3.

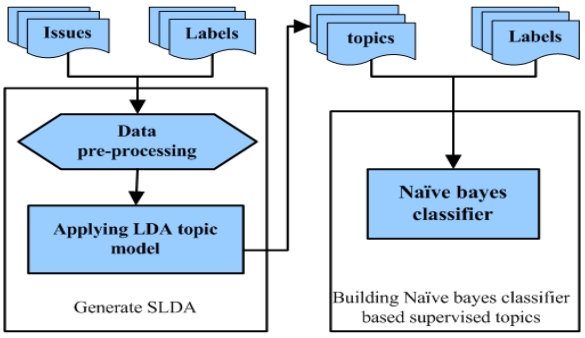

Fig. 3. After the process of the issue messages with single label, training the SLDA to generate the supervised topic distributions of PRs, use the distributions to build the Naive Bayes classifier.

A. SLDA in NBSL

In SLDA model, we can get the supervised topics. This phase basically consisits of two steps described as follows:

1) Data pre-processing: The datasets are extracted from six open source projects hosted on the Github with userID,title,description shown as TABLE I. There are a lot of noisy in PRs, bad punctuations, HTML links and so on. The process is mainly divided into six parts as follows:(1) Change the text words to lowercase. (2) Remove HTML tags and some bad punctuations from each documents. (3) Remove stopwords like a, the, is and that from documents, since they carry less meaningful contexts. (4) Tokenize, split the documents to a set of terms. (5) Stemming, the algorithm based on the Porter Stemming algorithm [5]. (6) Error spelling correction via the spelling correction algorithm based weighted contexts [3].

Table II displays the word set with labels after data preprocess. And we can note that the words are all meaningful.

2) Supervised topics generation after SLDA: To generate the supervised topics, first we extract the PRs with a label from each project, and then use LDA generate topics vector with the above dataset, and finally combine the label to each topics. In SLDA model, topics(K) is not the fix number, therefore, we try to find the most appropriate K by experiments.

B. Naive Bayes classifier based on supervised topics in NBSL

In the second phase of building NBSL, the objective is to build a classification model which can classify a PR message to a specify category. Based on the priori probability vector provided by the data processed after SLDA, we employ the Naive Bayes based on supervised topics. It is an important part of the NBSL to combine supervised topics model with Naive Bayes classifier. In this paper, topics vector generated by SLDA are used as attribute for Naive Bayes. The working process of naive Bayes classifier based topics is:

(1)given an input PR k, calculate the posterior probability between labels c, $p(Y = c|K = k)$;

$$p(Y = c|K = k) = \frac{P(Y = c)\pi_j P(K^{(j)} = k^i|Y = c)}{\sum_i P(Y = c)\pi_j P(K^j = k^j|Y = c)}, \quad k = 1, 2, ..., n; j = 1, 2, ..., n \quad (1)$$

(2)for the posterior probability distribution in Naive Bayes classifier, extract the class with largest probability value;

$$y = \arg\max P(Y = c)\pi_j P(K^j = k^j|Y = c) \quad (2)$$

As the formula(1) and formula(2) shows, k is stands for every PR, c is the category of PRs ,there are posterior probability vectors after the classification model, and the largest value provide a one-one relationship between labels and PRs which record the most closest topic message of PRs.

C. The algorithm of NBSL

The NBSL algorithm is summarized in the pseudo code. The general idea is to extract topic vectors with labels $k = (k_1, k_2, ..., k_n, c)$ in the training phase of SLDA as the input documents(D) in Naive Bayes, then return the prior

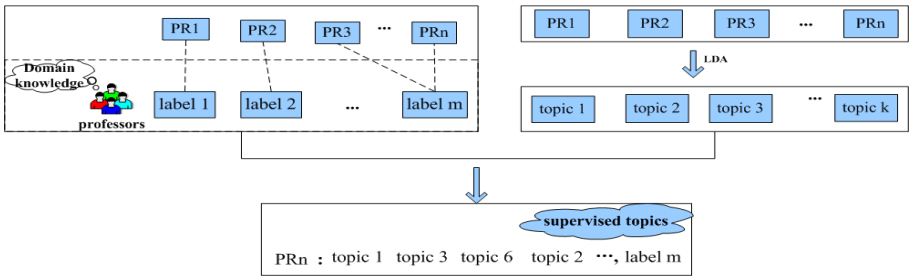

Fig. 2. Supervised topics generation after SLDA.

and conditional probability(condprob) to calculate posterior probability in classification.

TABLE I
THE ALGORITHM OF NBSL

The Algorithm of NBSL

Input: set of training samples after data processing
$X = x_1, x_2, ..., x_n$
1.initialize the parameters randomly:α, β
2.suppose all documents m[1,m] with labels
3. suppose all Topics $k \in [1, k]$
4. Topic vectors $\overrightarrow{\varphi k} \sim Dir(\overrightarrow{\beta})$
5. sample mixture proportion $\overrightarrow{\theta_m} \sim Dir(\overrightarrow{\alpha})$
6. sample document $lengthl_m \sim Poiss(\xi)$
7. for all words $n \in [1, N_m]$ in document m do
8. Topic Probability distributions $Z_{m,k} \sim Mult(\overrightarrow{\theta_m})$
9. Topic- word distribution $W_{m,n} \sim Mult\overrightarrow{\varphi Z_m}$
10.Interaction until astringent
11.Return α, β
12.For new documents
13. Repeat 3-10 with new , in 11
14.suppose Topics with label in $D = K_1, K_2, , , K_n$
15.$V \leftarrow extractedtopic(K)$
16.$m \leftarrow numberofdocuments(D)$
17.suppose each $c \in L$
18.N_c count docs in class(D,c)
19. $Prior[c] \leftarrow \frac{N_c}{m} N$
20. for all c in L do
21. $text_c \leftarrow consistofalldocumentsinclass(D, c)$
22. suppose $k \in V$
23. $T_{ck} \leftarrow counttimesofterm(text_c, k)$
24. for each $k \in V$ do
25. $condprob[k][c] \leftarrow P(K = k|Y = c) = \frac{t_{ck}+1}{\sum_{ck}(t_{ck}+1)}$
26.return V,prior,condprob
27.**END**

The cost to training the supervised classification is $O(N_{iter}N_D K\bar{l})$. Because of the training are divided into two parts: SLDA and Naive Bayes based on topics, the calculation belongs to two sections too. Relevant score of SLDA is $O(N_{iter}N_D K\bar{l})$, and Naive Bayes based on topics is $O(N_D\bar{l} + |C||V|)$. Thus the total cost for building NBSL is

$$O(N_{iter}N_D K\bar{l}) = O(N_{iter}N_D K\bar{l}) + O(N_D\bar{l} + |C||V|) \quad (3)$$

IV. EXPERIMENTS DESIGN AND RESULT ANALYSIS

A. Dataset

Six open source projects are chosen to build, test, and verify this method. The details of the projects are shown in TABLE III. The number of issues for eslint is close to 3000, which is different in languages with other projects. From PR mechanism we know PR is a special kind of issue, and a few labeled PRs are given in projects, but generally there are many labeled issues. So issues are considered as the training data. Further, we extract the issues with a single label, and employ N-fold cross-validation and split to do the experimental template.

TABLE II
DETAILS OF PROJECT

projects	issue	language
emberjs	9777	JavaScript
eslint	2863	JavaScript
riak	801	Shell
Automattic	443	PHP
facebookincubator	222	JavaScript
KnAllEdge	252	JavaScript

B. Comparison of NBSL to other algorithm

1) Topic-based classifiers: In this part, we apply the supervised topics model to the corpora of pre-processed issues of six projects with Naive Bayes, IBK(one algorithms of KNN), ID3(one algorithms of Decision Tree). Labels of all datasets are added by professionals, the accuracy of the labels is no doubt. And We have conducted experiments in the situation of different values of topics k=5,10,20, 50,100,150, respectively.

2) Topic-based vs Word-based classifiers: To build word-based classification model, we first extract the word vectors from each corpus, a corpus represents one of the project in Github, and then choose subsets of the features(words) to train the classification models with the same classifiers as the topic-based.

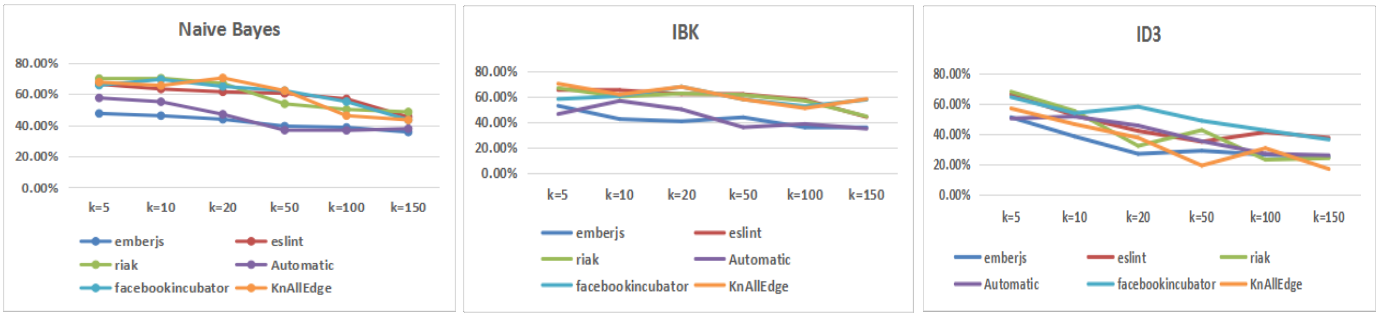

Fig. 4. Performance curves of six project with three classifiers.

C. Parameters of NBSL model experiments

1) Choosing the number of topics(K): In order to find out the most appropriate value of K. we conduct our comparison experiments on NBSL in the situation of K=5,10,20,50,100,150.

2) Choosing the number of top-N topics: Generally, not all of the topics extracted from the issues are helpful for classification, maybe only the most related topics of the issues are meaningful. So we will choose different numbers of most related topics , finding out how the precision of classification has changed with the changes of K and the number of most related topics.

3) Comparing the results in various project with same top-N: To explore whether the number of topics and the number of most related topics have influences on the performance of our model in all project, a comparison experiment is needed to be done in this work. In the experiment, different number of topics extracted with top-1 related topic to explore the effect in the project.

D. Evaluation of NBSL

The performance evaluation metric used to evaluate NBSL is precision score.The precision is calculated as:

$$precision(H,V) = \frac{1}{|V|}\sum_{i=1}^{|V|}\frac{|Y_i \bigcap Z_i|}{|Z_i|} \tag{4}$$

where V is our validation set, H indicates our classifier, indicates the prediction label set of message , indicates the label set by participants which added by professors. In our experiment, we employ N-fold cross-validation and split to validate the model. The idea of N-fold cross-validation is that the dataset is randomly partitioned into N parts, one part is retained as a testing data and the other N-1 parts are used as a training data. We report the average value after N runs of cross-validation. And the split is that splitting the dataset into two parts with appropriate proportion.

E. Experiment Results and Analysis

1) Comparisons results: ***Comparison of topic-based classification models.*** The Comparison results shown in Fig.4. The X axis represents the number of topics extracting from each issue, the Y axis represent the precision of classification. Note that the supervised topics model with the Naive Bayes classifier(NBSL) yields better results in these projects. When consider on trends of the performance,Naive Bayes model is further developed with a higher probability.

Comparison between the word-based and topic-based classification models. we compare the classifying performance between topic-based and word-based model, as shown in TABLE V, considering the average length of PRs, we adopt a feature selection algorithm to select feature words with a number of 5,10,20. For topic-based models, we collect the results from the models trained from 5,10 topics since they are likely to produce significantly high performance for all cases with less required time as shown in TABLE IV. In most cases,topics-based model slightly outperforms word-based model.

2) NBSL experiments result analysis: We analyze different number of topics and top-N associated topics with the issues in order to get a better result for PR classification. As shown in table III, projects Emberjs and Eslint have more than 2,000 issues, and we treat them as the large scale project; project Riak has more 800 issues, but less than 2,000, and we treat them as medium scale project; Projects Automattic, KnAllEdge, and Facebookincubator have less than 500 issues, and we treat them as small scale project.

Impact of various number of topics on precision. As shown in the first picture in Fig.4, the projects we chose when the value of K is set to 5,10, the precision of classification of the projects is better than that of the others on the whole. On average, the model can reach a precision of 60%. We note that the more topics have extracted, the lower accuracy is our classification, which is the emergence of over-fitting. So the selection of the value of K is very important for the PR classification.

Numbers of selected top-N analysis. To explore the effect of using the most related topics for classification, we chose different number of most related topics with different K to training and classifying, calculating the classification precision of each one, the results is shown in Fig.5. As shown in Fig.5, whatever the value of K is set to, the approach using the most related topics of the issue always perform better than that using the all topics, and it can be concluded that the approach using the top-1 related topic can perform better than that in all other situations. For a issue, it always try to present one theme, and the label of the issue is the closest to the theme. Meanwhile,

TABLE III
DETAILS OF PROJECT

topics(K)	Naive Bayes			IBK			ID3		
	riak	facebookincubator	KnAllEdge	riak	facebookincubator	KnAllEdge	riak	facebookincubator	KnAllEdge
5	69.77%	65.52%	67.44%	66.28%	57.69%	69.77%	67.65%	64.10%	56.67%
10	70%	69.23%	65.31%	59.60%	60%	61.22%	55%	53.54%	46.30%
20	66.67%	64.71%	70.18%	62%	67.33%	67.33%	32%	57.78%	37.38%
50	53.57%	62%	62%	60.87%	57.43%	57.43%	42.39%	48.51%	35%
100	50%	55%	46%	56.25%	52%	50.50%	22.92%	42.22%	31.60%
150	48.39%	43.27%	43.27%	44%	57.14%	57.69%	24%	36.17%	27.50%

TABLE IV
DETAILS OF PROJECT

	Selected Features	Naive Bayes			IBK			ID3		
		riak	facebookincubator	KnAllEdge	riak	facebookincubator	KnAllEdge	riak	facebookincubator	KnAllEdge
word based	5	65.57%	60.66%	59.32%	66.00%	59.77%	61.02%	64.71%	59.77%	61.02%
	10	64%	62.30%	59.32%	61.00%	60%	55.17%	66%	59.77%	60.34%
	20	63.93%	57.14%	52.46%	61.00%	61.02%	60.92%	61%	61.02%	59.77%
topic based	K=5	69.77%	65.52%	67.44%	66.28%	57.69%	69.77%	67.65%	64.10%	56.67%
	K=10	70%	69.23%	65.31%	59.60%	60%	61.22%	55%	53.54%	46.30%

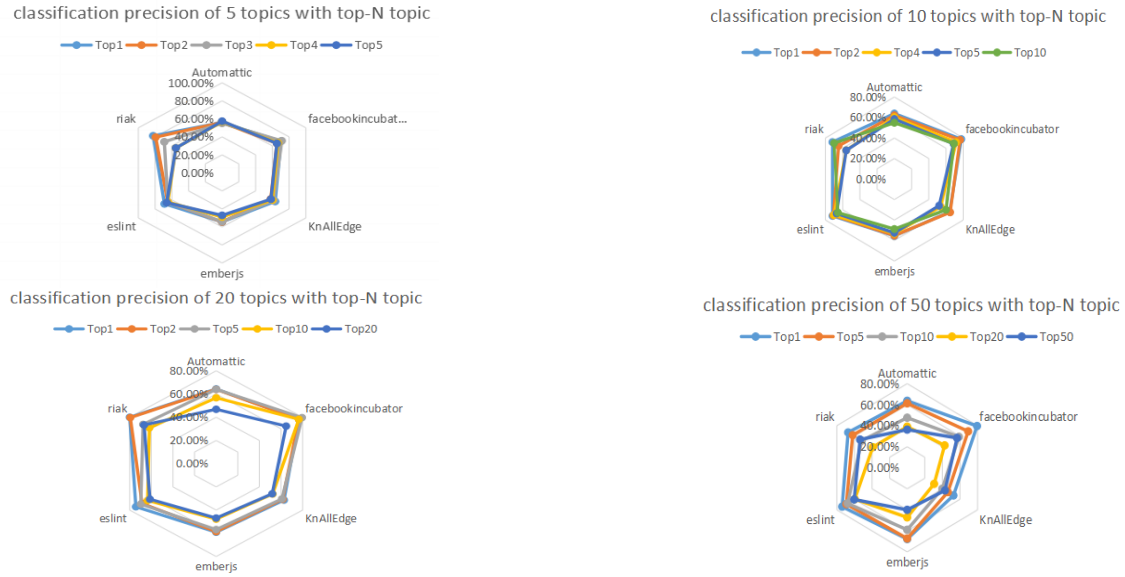

Fig. 5. Classification precision of different number of topics with top-N topic in various projects

the top-1 related topic of the issue is also always represents the theme of issue closely. Thus, it can benefit the classification better.

Analyze the efficiency using top-1 topic of different number of topics. From the above experiments, we find that the number of most related topics are two contribution factors that can influence the performance of the model. After comparing the precision of different number of topics extracted with top-1 related topic, we note that a small K will perform better for small scale project using top-1 related topic, but a big K will preform better for a medium or large project.The details are shown in Fig.6. We can see that the number of topics is almost directly proportional to the size of the project. The larger amount of data, the more number of topics, the better meaning of the text responded. The reason for this phenomenon is that, the larger the project, the more problems will arise, it trend to generate more category labels at the same time, and thus we need generate more topics to fit them, but for a small project

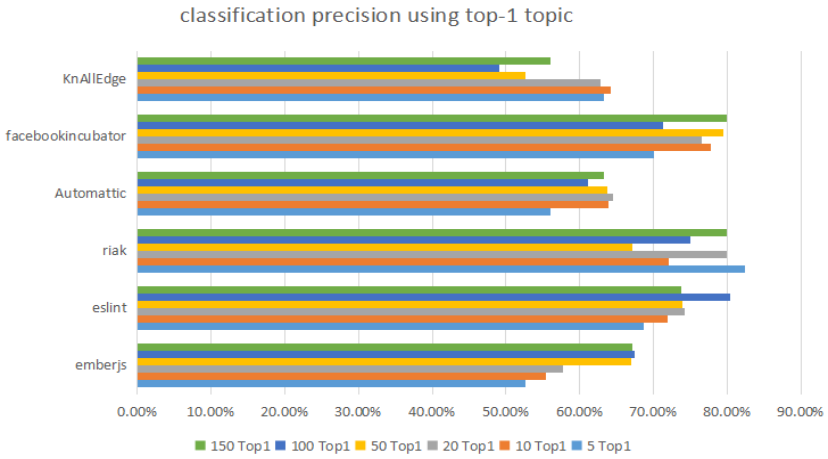

Fig. 6. Classification precision of using top-1 topic of topic distribution in six projects.

owning a little number of labels, extracting more topics will cause the over-fit problems.

V. Conclusions and Future Work

In order to improve the efficiency processing of PR, we propose NBSL classification model to sort PRs. After demonstrating the performance of the propose model on six projects hosted on Github, We note when using SLDA to generate the supervised topic distributions, the number of topics K=5,10 can get a better result in experiments. On average, the classification model can reach a precision of 60%. We also find using the top-N related topics of topic distribution to classifying PRs is helpful in improving the precision. In a majority of situations, using top-1 can provide a better result, and a small K will helpful for small scale project, a big K will helpful for medium and large scale project.

However, there still exist a lot of problems to be resolved. For example, some PRs are related to many modules which requires a more effective way to add multi-category labels to them. It can be observed from the experimental data that some PRs are very short, even just a word, null value phenomenon shows up after the data processing which has an effect on the classification result seriously. These problems are still waiting for solutions in the future.

Acknowledgment

This work is supported by the Fundamental Research Funds for Central Universities of the Central South University(No.2017zzts600) and Chile Conicyt FONDECYT Regular 2018, ”Big data for sustainable smart cities with the aids of computational. intelligence”

References

[1] Antoniol, G., Ayari, K., Penta, M.D., Khomh, F.: Is it a bug or an enhancement?:a text-based approach to classify change requests. In: Conference of the Centre for Advanced Studies on Collaborative Research, October 27-30, 2008, Richmond Hill, Ontario, Canada. p. 23 (2008)

[2] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)

[3] Dey, L., Haque, S.M.: Opinion mining from noisy text data. International Journal on Document Analysis and Recognition (IJDAR) 12(3), 205–226 (2009)

[4] Fu, Y., Yan, M., Zhang, X., Xu, L., Yang, D., Kymer, J.D.: Automated classification of software change messages by semi-supervised latent dirichlet allocation. Information and Software Technology 57(1), 369–377 (2014)

[5] Mccallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION 62(2), 41–48 (1998)

[6] Pingclasai, N., Hata, H., Matsumoto, K.I.: Classifying bug reports to bugs and other requests using topic modeling. In: Asia-Pacific Software Engineering Conference. pp. 13–18 (2013)

[7] Rahman, M.M., Roy, C.K.: An insight into the pull requests of github. In: Working Conference on Mining Software Repositories. pp. 364–367 (2014)

[8] Willett, Peter: Readings in information retrieval /. Morgan Kaufman, (1997)

[9] Yu, Y., Wang, H., Filkov, V., Devanbu, P., Vasilescu, B.: Wait for it:determinants of pull request evaluation latency on github. In: Ieee/acm Working Conference on Mining Software Repositories. pp. 367–371 (2015)

[10] Yu, Y., Wang, H., Yin, G., Ling, C.X.: Reviewer recommender of pull-requests in github. In: IEEE International Conference on Software Maintenance and Evolution. pp. 609–612 (2014)

[11] Zhang, Y., Yin, G., Yu, Y., Wang, H.: A exploratory study of @-mention in github’s pull-requests. In: Asia-Pacific Software Engineering Conference. pp. 343–350 (2014)