Issue report classification using a multimodal deep learning technique

Changwon Kwak [1] and Seonah Lee [1,2]

[1] Department of AI Convergence Engineering, Gyeongsang National University, Jinju, South Korea
[2] Department of Aerospace and Software Engineering, Gyeongsang National University, Jinju, South Korea

Abstract

Issue reports are useful resources for developing open-source software and continuously maintaining software products. However, it is not easy to systematically classify the issue reports accumulated hundreds of cases a day. To this end, researchers have studied how to classify issue reports automatically. However, these approaches are limited to applying a text-oriented classification method. In this paper, we apply a multi-modal model-based classification method, which has shown great performance improvement in many fields. We use images attached to an issue report to improve the performance of issue report classification. To evaluate our approach, we conduct an experiment, where we compare the performance of a text-based single-modal model and that of a text and image-based multi-modal model. The experimental results show that the multi-modal method yields 2.1% higher classification f1-score than that of the single-modal method. Based on the experimental results, we will continue our further exploration of the multi-modal model, by considering the characteristics of the issue report and various heterogeneous outputs.

Keywords

Multimodal Deep learning, Classification, Issue reports

1. Introduction

Today, when developing and continuously maintaining open-source software, open-source contributors use issue management systems as a way to quickly reflect users' inconveniences and improvements of the software systems. Stakeholders report bugs, functional improvements, and other requests they find while using the software as issues. Developers refer to the issue report to discuss and improve the software. In the case of active open-source projects, these issue reports are generated and accumulated by hundreds of cases per day. In such a situation, it is not easy to systematically classify and manage issues.

Researchers have proposed automatically classifying issue reports to manage them more systematically [1,2,3,4,6,7]. Recently, researchers began to adopt deep learning techniques to classify issue reports. For instance, Cho et al. [8] used CNN and RNN deep learning techniques to classify issue reports. However, existing approaches obtain text data such as titles and body contents of issue reports as inputs for training their models. Those approaches do not use various kinds of information that issue reports include.

Meanwhile, in the area of deep learning techniques, multi-modal deep learning models using two or more modalities have shown significant performance improvement in many fields [9,10,11,12,13]. This shows that we could achieve better performance by overcoming the limitations of using only single-modal data.

We observed that issue reports often contain relevant images. We, therefore, decided to apply a multi-modal model-based classification method to classify issue reports. Our proposed method classifies issue reports by combining the representation of text data and image data of issue reports based on the method of Antol, Stanislaw, et al. [9]. We also conducted an experiment to see whether our approach could achieve higher performance. To evaluate our multi-modal model-based approach, we compare the performance of

1st International Workshop on Intelligent Software Engineering, December 06, 2022, Busan, South Korea
EMAIL: chang_26@naver.com (A. 1); saleese@gnu.ac.kr (A. 2)

our approach with that of the CNN-based model method of a single-modal model, Cho et al. [8]. For this, we collected the issue reports of Vscode, a major project of GitHub. We finally collected 17,500 issue reports with one or more images. To resolve data imbalance issues, we downsampled issue reports and used 8,500 issue reports. As a result of the experiment, our approach showed an improved f1-score of about 2.1%, compared to the classification model of the existing method [8].

The paper is organized as follows. Section 2 introduces related works. Section 3 explains the experimental setup. Section 4 presents the experimental results. Section 5 discusses the experimental results and Section 6 concludes.

2. Related work

The related studies to ours are the studies that classified the issue reports of open-source projects and the studies that applied multi-modal deep learning.

First, there are attempts to conduct the binary classification of issue reports into bugs/non-bugs. For example, Pandey et al. [1] extracted a summary from an issue report and classified the issue report as a bug/non-bug using Naive Bayes and SVM. In addition, Zhu et al. [2] used kNN to determine whether the existing label is correct, and classified an issue report as bug/non-bug using Attention-based Bi-directional LSTM. As the next multi-class classification, Kallis et al. [5] used FastText to classify an issue report into Bug, Enhancement, or Question. Kochhar et al. [6] classified issue reports into 13 categories including BUG, using SVM. Also, Fazayeli et al. [7] tried to classify issue reports into five categories: unclear, question, up for grabs, bug, and others, and used the SMO machine learning algorithm. Recently, Cho et al. [8] proposed a method of classifying issue reports into features of the software using a user manual with CNN and RNN (i.e. LSTM) deep learning techniques. In this paper, we conduct a comparative experiment with the CNN model of Cho et al. [8] as the baseline.

Researchers widely used multi-modal deep learning models in the fields of Action Recognition, Image Generation, Image Captioning, and Visual Question Answering (VQA). Antol, Stanislaw, et al. [9] showed good performance of a multi-modal model in VQA work using a model with two channels, image and text(question). Antol, Stanislaw, et al. [9] used VGGNet for image channels and LSTM for text channels to embed each data. Their proposed approach combines features through element-wise multiplication to transform the data into a common space to make a classification. Although there are more effective methods such as MUTAN [13], MCB [11], and MLB [12] as data combining methods, this work uses element-wise multiplexing from Antol, Stanislaw, et al. [9] to reduce model operations and simplify implementation. That is, we experiment with whether a multi-modal deep learning model extracting text and image data from issue reports can improve the performance of issue classification, and we report the results.

3. Experimental set-up

3.1. Dataset

Table 1
Number of issues for labels

DataSet	Label		Total
	Bug	*Feature*	
Total	13,507 (77.2%)	3,988 (22.8%)	17,495
DownSampling	4,500 (53%)	3,988 (47%)	8,488

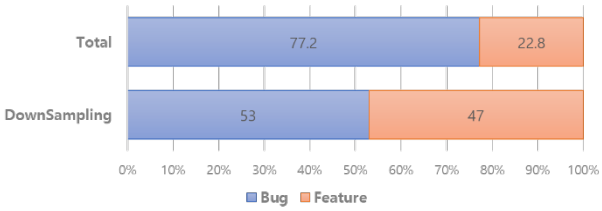

Figure 1: Ratio of issues for labels

We used the open-source project Vscode, a major project of GitHub, in our experiment. Among the issue reports of Vscode, we collected issue reports with more than one image. We collected the issue reports that were labeled ‘bug’ or ‘feature’. The total number of collected data was 17,500, and we used the first image that is most closely related to the issue among the images of each data. Figure 1 shows the ratio of the collected issue data for label. Finally, we used about 8,500 issue reports through the DownSampling method to resolve the imbalance of data with the different numbers of data for each label and to speed up the experiment by reducing the model size. We used all of the 'Feature' label data, which are relatively little data, and for the

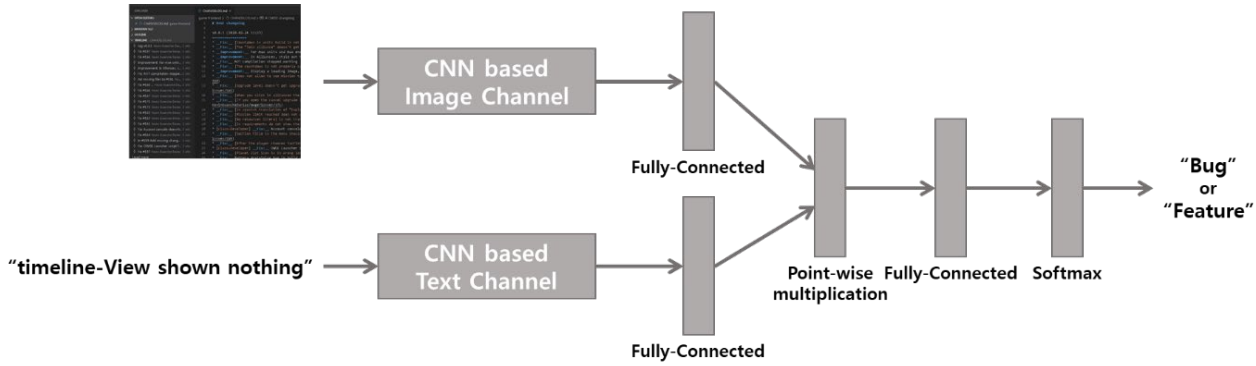

Figure 2: The structure of the proposed model

'Bug' label data used an appropriate amount of data from the latest data. So, we used all 'feature' label data and 4,500 'bug' label data from the latest data.

3.2. Method

Figure 2 shows the structure of the proposed model classifying the issue report using the multi-modal model. As shown in Figure 2, the model gets the text (title) data and image data of issue reports as inputs. The model classifies the issue report into "bug" or "feature.". The model passes the image and text data through a CNN-based channel, respectively, to extract expression vectors. These features are combined through point-wise multiplication operation to express them in a common space. After that, the model performs a softmax operation and finally makes a classification of the issue report as an output.

3.3. Measurements

The metrics used for measuring classification performance were precision, recall, and f1-score. The calculation for each metric was conducted using the equations below.

$$Precision = \frac{\sum_{i=0}^{n}(precision_i * number\ of\ class_i's\ issue\ reports)}{\sum_{i=0}^{n}(number\ of\ class_i's\ issue\ reports)} \quad (1)$$

$$precision_i = \frac{tp}{tp+fp} \quad (2)$$

$$Recall = \frac{\sum_{i=0}^{n}(recall_i * number\ of\ class_i's\ issue\ reports)}{\sum_{i=0}^{n}(number\ of\ class_i's\ issue\ reports)} \quad (3)$$

$$recall_i = \frac{tp}{tp+fn} \quad (4)$$

$$F1 - score = \frac{\sum_{i=0}^{n}(f1{-}score_i * number\ of\ class_i's\ issue\ reports)}{\sum_{i=0}^{n}(number\ of\ class_i's\ issue\ reports)} \quad (5)$$

$$f1 - score_i = 2 * \frac{precision_i * recall_i}{precision_i + recall_i} \quad (6)$$

In the above equations, tp represents the number of issue reports that the model predicted to $class_i$ that belonged to $class_i$. The symbol fp represents the number of issue reports that the model predicted to $class_i$ but did not belong to $class_i$. The symbol tn denotes the number of issue reports that the model predicted to not $class_i$ and did not belong to $class_i$, and fn denotes the number of issue reports that the model predicted to not belong to $class_i$ but belonged to $class_i$.

4. Experimental results

Table 2 shows the performance differences between the proposed classification model and the issue report classification of the existing classification model. The performance metrics are precision, recall, and f1-score, and the metric of each class is calculated and a weighted average is used according to the class frequency. The results of the proposed model, the multi-modal model, showed 73.432% precision, 73.460% recall, and 73.423% f1-score. The CNN model, which is a single-modal model, showed 71.356% precision, 71.386% recall, and 71.315% f1-score. As a result, the proposed classification model performed better than the existing classification model.

However, it is difficult to regard that it as a meaningful improvement because the performance improvement is insignificant. To determine whether the image data used in the experiment is suitable for the classification task, we constructed a single-modal model using only the image data to measure the classification

accuracy. As a result, the single-modal model showed 53.373% precision, 54.250% recall, and 53.467% f1-score, which does not seem to help the image data with the classification task.

Table 2
Result comparison between singlemodal and our multimodal model

MODEL	Precision	Recall	F1-score
CNN (Text Only)	71.356	71.386	71.315
CNN (Image Only)	53.373	54.250	53.467
Multimodal (Text+Image)	73.432	73.460	73.423

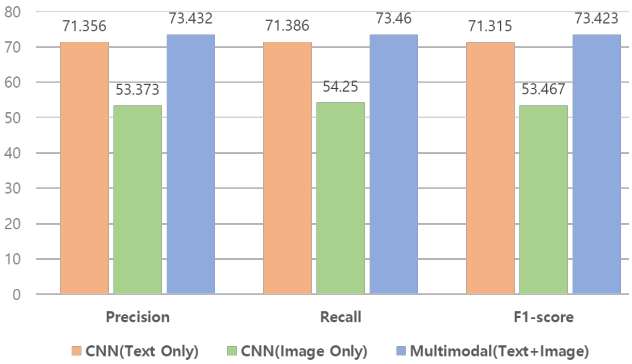

Figure 3: Result comparison between single-modal and our multimodal model

5. Discussion

Existing issue report classification studies have limitations in applying text-oriented single-modal classification methods such as title and body content. In addition to text modality, there are other modality data in issue reports. In particular, we conducted this study based on the fact that images exist in many issue reports. However, the results of our proposed model did not improve the performance as expected. Therefore, we conducted an additional experiment and checked the accuracy of a deep learning model that only uses images of issue reports. The classification accuracy of the deep learning model using only images is around 53.5 % f1-score. This means that the image data used in this study are not primary factors on classification performance. Even so, the information that images have is helpful for classification work. In fact, it is easy for developers to understand the issue report when they see the text and image data of the issue report together. Most of the images in the body of the issue report are parts of code captured in the development environment. Compared to using the source code directly, it seems complicated to understand the meaning of data in image form. Therefore, it is quite difficult to distinguish the differences between the issue reports labeled "bug" and the issue reports labeled "feature". Now, we question if we recognize the source code from images, the source information will be able to help our classification.

Nonetheless, based on these experimental results, we were able to confirm the effect of the multi-modal application of the issue report. Therefore, we will continue our further exploration of the multi-modal model, which takes into account the characteristics of the issue report and various heterogeneous outputs. First, most of the images attached to issue reports contain code and text. Therefore, if we extract the code and text from the image and use them for classifying, it is expected to show better performance than the existing method using the image. Next, since users can attach codes to the issue report, we can use the code as another modality. Since the code is a source that is directly related to the software issue, it is highly valuable. Therefore, it is possible to try to improve the performance by using it as a multi-modal together with the existing text data.

6. Conclusion

We have proposed a method for classifying issue reports based on a multi-modal deep learning model using text data (title) and image data (body) of the issue report. Experimental results show that the classification model of the proposed method has an f1-score improvement of about 2.1% over the existing classification model, and that the multi-modal deep learning model is positive for improving the performance of the classification task.

We infer that these results come from the fact that the model utilizes various information from the issue report. When users write an issue report, they often write a description of the issue by attaching images, videos, and codes, etc., in addition to the title and body content in text format. This is actually very helpful data for humans to understand. Therefore, we infer that the model could better represent the issue report when

also using images that are directly related to the content rather than just the text of the title or body, resulting in better classification performance. In the future, we will explore and advance the utilization strategy of image data in issue reports. And we will create a multi-modal model that uses more heterogeneous components of issue reports for more accurate issue classification.

7. References

[1] N. Pandey, A. Hudait, D.K. Sanyal, A. Sen, Automated classification of issue reports from a software issue tracker. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Springer, Singapore, pp. 423–430, 2018.

[2] Zhu, Y., Pan, M., Pei, Y., & Zhang, T. "A Bug or a Suggestion? An Automatic Way to Label Issues.". arXiv:1909.00934, 2019.

[3] A. Panichella, A systematic comparison of search algorithms for topic modelling—a study on duplicate bug report identification, in: International Symposium on Search Based Software Engineering, Springer, Cham, pp. 11–26, 2019.

[4] M. Lu, P. Liang, Automatic classification of non-functional requirements from augmented app user reviews, in: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, pp. 344–353, 2017.

[5] R. Kallis, A. Di Sorbo, G. Canfora, Panichella, Ticket Tagger: machine learning driven issue classification, in: 2019 IEEE International Conference on Software Maintenance and Evolution(ICSME), IEEE, pp. 406–409, 2019.

[6] B. Wang, R. Peng, Y. Li, H. Lai, Z. Wang, Requirements traceability technologies and technology transfer decision support: A systematic review, Journal of Systems and Software, 2018.

[7] R. White, J. Krinke, R. Tan, Establishing multilevel test-to-code traceability links, in: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, 2020.

[8] CHO, H., LEE, S., KANG, S., Classifying issue reports according to feature descriptions in a user manual based on a deep learning model. Information and Software Technology, 142: 106743, 2022.

[9] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 2425-2433.

[10] KAFLE, Kushal; KANAN, Christopher. Visual question answering: Datasets, algorithms, and future challenges. Computer Vision and Image Understanding, 2017, 163: 3-20.

[11] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., & Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847, 2016.

[12] Kim, J. H., On, K. W., Lim, W., Kim, J., Ha, J. W., & Zhang, B. T. Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv:1610.04325, 2016.

[13] Ben-Younes, H., Cadene, R., Cord, M., & Thome, N. Mutan: Multimodal tucker fusion for visual question answering. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 2612-2620.