Multi-label commit message classification through p-tuning

Xia Li, Tanvi Mistry

The Department of Software Engineering and Game Design and Development,
Kennesaw State University,
Marietta, USA

Abstract. Version control systems (VCS) play a crucial role by enabling developers to record changes, revert to previous versions, and coordinate work across distributed teams. In version control systems (e.g., GitHub), commit message serves as concise descriptions of code changes made during development. In our study, we evaluate the performance of multi-label commit message classification using p-tuning (learnable prompt templates) through three pre-trained models such as BERT, RoBERTa and DistilBERT. The experimental results demonstrate that RoBERTa model outperforms other two models in terms of the widely used evaluation metrics (e.g., achieving 81.99% F1 score).

Keywords: Multi-label commit message classification, p-tuning, pre-trained models

1 Introduction

In modern software development, version control systems (VCS) play a crucial role by enabling developers to record changes, revert to previous versions, and coordinate work across distributed teams. By leveraging VCS, organizations can ensure code integrity, streamline development workflows, and maintain a comprehensive history of code changes. Among the various VCS tools, Git has been widely adopted due to its distributed nature, flexibility, and efficiency in handling large-scale projects. GitHub[1], built around Git, is one of the largest platforms for version control and source code management, with a vast user base of over 50 million developers worldwide. GitHub serves as a central hub for open-source and enterprise-level software development, facilitating collaboration among developers through features like pull requests, issue tracking, and project management tools. One benefit of GitHub is that it can provide robust APIs that grant access to extensive code repositories, offering valuable insights into software development trends, best practices, and industry-wide collaboration patterns.

In version control systems (e.g., GitHub), commit message serves as concise descriptions of code changes made during development. These messages help developers understand the purpose of modifications, track feature updates, and diagnose

[1] https://github.com/

David C. Wyld et al. (Eds): NLMLT, SEAU, ITCS, CST – 2025
pp. 81-88, 2025. CS & IT - CSCP 2025

DOI: 10.5121/csit.2025.151608

issues efficiently. Well-structured commit messages facilitate seamless collaboration, reduce debugging time, and contribute to better software maintenance. However, developers often write commit messages in informal or inconsistent formats, making it challenging to systematically analyze them. Accurate classification of commit messages can provide valuable insights into software evolution, development patterns, and code quality.

Many researchers utilize Natural Language Processing (NLP) and Neural Networks (NN) for commit message classification. For example, Mariano et al. [5] propose to use XGBoost to improve commit classification. Meng et al. [1] propose to classify commit messages by capturing various syntactic and semantic relationships between co-applied changes via convolutional neural networks (CNN). Recently, pre-trained models (e.g., BERT [4], GPT [6]), which are trained on a large dataset in advance, have been widely used for many downstream tasks in various AI fields such as natural language processing (NLP) and computer vision (CV). Pre-trained models are also applied into the field of software engineering. For example, Sarwar et al. [2] present a multi-label commit message classification technique based on bidirectional neural networks DistilBERT [16] with transfer learning. Researchers also combine the prompt engineering and pre-trained models by appending various prompts after the input sequence and the target task is masked so that pre-trained models can predict the masked label [8], showing more promising performance than techniques with stand-alone pre-trained models. In this paper, we conduct an extensive study to evaluate the performance of multi-label commit message classification by applying p-tuning [10] which is a prompt-based approach on various pre-trained models such as BERT [4], RoBERTa [9] and DistilBERT [16]. The intuition to use p-tuning is that it can provide stable performance for classification tasks through learnable and flexible templates [10]. In our study, we propose to convert the task of multi-label commit message classification into several binary classification tasks, based on the number of labels in the dataset. The results in our study demonstrate that RoBERTa model outperforms other two models in terms of the widely used evaluation metrics (e.g., achieving 81.99% F1 score).

The structure of the paper is as follows. In Section 2, we introduce various studies related to commit message classification. In Section 3, we introduce the studied approach in our paper. In Section 4 and Section 5, we discuss our experimental design and results analysis, respectively. We discuss the threats to validity in Section 6 and conclude the paper in Section 7.

2 Related Work

In this section, we discuss related studies of commit message classification through traditional machine learning/deep learning and pre-trained models.

2.1 Commit classification via traditional learning techniques

Commit classification has been extensively explored using traditional machine learning as well as deep learning methods. For example, Santos. et al. [12] explored the application of natural language processing (NLP) techniques to classify software commits based solely on commit messages by applying traditional machine learning models such as Naive Bayes, Random Forests, and SVMs. Mariano et al. [5] proposed to use XGBoost to improve commit classification. Meng et al. [1] proposed to classify commit messages by capturing various syntactic and semantic relationships between co-applied changes via convolutional neural networks (CNN). Wu et al. [11] employed a BiLSTM model to identify security-related patches by modeling the structural dependencies in commits, significantly improving classification performance.

2.2 Commit message classification via Pre-trained models

Inspired by the theory of transfer learning, researchers seek to apply powerful pre-trained models into the filed of commit message classification. For example, Sarwar et al. [2] presented a multi-label commit message classification technique based on bidirectional neural networks DistilBERT [16] with transfer learning. Ghadhab et al. [13] proposed to use BERT model for the classification of commits into three categories of maintenance tasks by better understanding the context of each word in the commit message. Heričko et al. [14] extracted semantic features from commits based on modifications in the source code and used two BERT-based code models (CodeBERT and GraphCodeBERT) to improve commit message classification. Zeng et al. [15] compared code changes at the hunk level, took fine-grained features based on categories of changed files, and aggregated with the representation of commit messages to improve the classification based on ChatGPT.

In this paper, we conduct an extensive study on the impact of multi-label commit message classification by fine-tuning three pre-trained models (BERT, RoBERTa and DistilBERT) and applying p-tuning, which is a popular prompt-based learning technique by designing learnable templates that can be adapted into the models for training.

3 Study Approach

Figure 1 shows the general process of our study. In following subsections, we demonstrate the key approaches in our study including p-tuning (Section 3.1), data pre-processing specifically for multi-label commit message classification (Section 3.2).

3.1 P-tuning

P-tuning [10] is a prompt-based learning technique that utilizes learnable continuous prompts to guide pre-trained language models in various classification tasks.

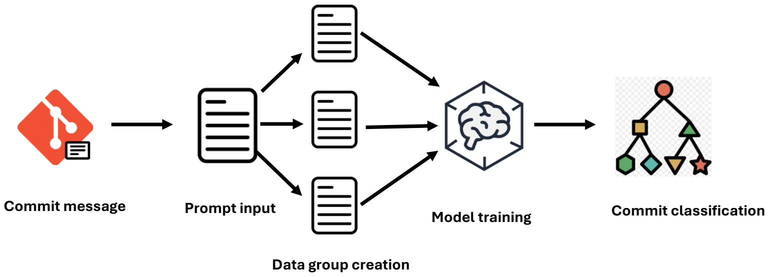

Fig. 1. Study process

It is inspired by an earlier technique called Pattern-Exploiting Training (PET) [8]. Unlike traditional fine-tuning methods that use pre-trained models to connect additional neural networks (e.g., RNNs or CNNs) for downstream tasks, the two techniques add the classification target label directly into the original text as a new prompt-style input. This approach strengthens the alignment between the input and the prediction target, enabling pre-trained models to directly predict masked tokens and improving the classification performance. The two techniques have shown strong performance in general NLP domains and also the field of software engineering [7]. However, PET only utilizes fixed prompts that are discrete and sensitive resulting in unstable or inconsistent performance, highlighted in studies such as prompt engineering, where the performances of models like BERT and GPT are significantly different with various prompt formulations. In comparison, p-tuning uses flexible templates by inserting a varying number of learnable tokens, either before or after the original text, serving as continuous prompts that can be trained by pre-trained models. However, these tokens are semantically ambiguous, making them harder for the model to learn effectively in isolation. In our study, we concatenate learnable tokens with manually designed prompt phrases to provide both semantic guidance and adaptability. The template as the input of the pre-trained models is as follows: *[CLS][P][P][P]Commit Message Statement[SEP]. [CLS][P][P][P]This message is related to [M][SEP].* In the template, [P] represents the learnable token while [M] is the masked token to represent the commit category (e.g., corrective, adaptive or perfective). [CLS] is a special token in the front of the original input text and [SEP] is a separator token to indicate the segment of each sentence.

3.2 Data pre-processing

In our study, we utilize the dataset developed by Sarwar et al. [2], which consists of 2037 labeled commit messages across different projects and languages extracted from GitHub including three different categories (i.e., corrective, adaptive and perfective). Prior to model input, we pre-process the data using standard natural language processing techniques such as stemming, lemmatization, stopword removal, and lowercasing via the widely adopted NLTK toolkit[2]. Based on the processed data, we construct a variety of prompts to feed into pre-trained models for commit message classification training.

In p-tuning, the pre-trained model models can predict only one label for a single training task while our study is related to multi-label commit message classification. To enable more effective modeling, we transform the original multi-label classification problem into a set of binary classification tasks based on the total number of categories in the dataset (e.g., three categories in our study). This approach allows standard classification techniques to be applied to each label individually, rather than attempting to predict multiple labels simultaneously to overcome the limitation of pre-trained models. The transformation process involves three main steps: (1) Group creation. We identify all unique labels in the dataset and create N separate groups, where N represents the total number of labels. Each group corresponding to a single label and contains the full dataset but treats this specific label as the target class. (2) Label assignment. In each group, each data point is re-labeled as either positive or negative, depending on whether it belongs to the label associated with that group based on the original dataset. For instance, if a data point originally has labels A and C, it would be labeled as positive in group A and group C, but negative in all other groups. (3) Classifier training. We then train N binary classifiers based on the groups. Each classifier learns to predict whether the input belongs to its associated label.

4 Experimental Design

In this study, we use three foundation pre-trained models BERT [4], RoBERTa [9] and DistilBERT [16] that can be downloaded from the popular AI community Hugging Face[3]. RoBERTa builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. DistilBERT is a light variant of BERT model with fewer parameters. For each data group, we split the dataset into a training set (80%) and a test set (20%). We perform 10-fold cross-validation for all three pre-trained models. Since we convert to binary classification task for each data group based on the Section 3.2, we use the binary cross-entropy loss function. To evaluate performance, we employ widely-used metrics: precision, recall, and F1 score, which are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

[2] https://www.nltk.org/

[3] Hugging Face. https://huggingface.co/

Where TP denotes the number of true positives, FP represents false positives, and FN indicates false negatives.

Since we train the three data groups independently, we use major different parameters as follows (from group 1 to group 3): we set the maximum input sequence length to 128, 128, 256, batch size to 8 for all groups, learning rate to 3×10^{-5} for all groups, and training epochs for 16, 16, 32. All models are optimized using the AdamW optimizer [17]. All training and inference are conducted on a server equipped with an Intel Core i9-13900K CPU, 32GB RAM, and an NVIDIA RTX 4090 GPU.

Table 1. Effectiveness of multi-label commit message classification

		BERT	**RoBERTa**	**DistilBERT]**
Precision	Group1	80.23%	81.04%	80.35%
	Group2	81.47%	82.92%	81.72%
	Group3	80.95%	81.88%	80.81%
	Average	80.88%	81.95%	80.96%
Recall	Group1	81.16%	82.01%	81.04%
	Group2	80.62%	81.75%	80.19%
	Group3	81.88%	82.37%	81.59%
	Average	81.22%	82.04%	80.94%
F1 score	Group1	80.69%	81.52%	80.69%
	Group2	81.04%	82.33%	80.95%
	Group3	81.41%	82.12%	81.20%
	Average	81.05%	81.99%	80.95%

5 Results Analysis

In this section, we investigate the performance of multi-label commit message classification through the three studied pre-trained models based on the flexible template introduced in Section 3.1. Table 1 shows the results of precision, recall, and F1 score in each data group representing three labels in the dataset. We also include the average values for the evaluation metrics. From the performance comparison across the models BERT, RoBERTa, and DistilBERT on three groups, we can find that RoBERTa consistently outperforms the other two models for all evaluation metrics. For example, on average, RoBERTa achieves the highest values: 81.95% Precision, 82.04% Recall, and 81.99% F1 score, indicating its robustness and superior generalization across different data groups/labels. The possible reason is that RoBERTa is trained on a much larger dataset (160GB vs. 16GB in BERT) and uses dynamic masking (masking different tokens during each training epoch) with longer training periods and larger batch sizes so that it can understand context more effectively. Another finding is that BERT performs comparable to DistilBERT in terms of the evaluation metrics studied. This finding indicates that DistilBERT is more practical in prompt learning tasks than BERT, since it requires fewer computational resources for training and inference compared to BERT.

During the evaluations above, we set different labels as independent groups while it is important to evaluate the effectiveness of multi-label commit message classification based on the three labels together. Thus, we use the following formula to represent the classification accuracy:

$$\text{Accuracy} = \frac{\text{Data points with correct predictions}}{\text{All data points}}$$

where "correct prediction" indicates that all labels are predicted correctly simultaneously based on the three independent classifiers. Figure 2 shows the accuracy results for the three pre-trained models. From the results, we can conclude that RoBERTa (e.g., 80.23% accuracy) still outperforms other two models.

Table 2. Performance of classfication accuracy

	BERT	**RoBERTa**	**DistilBERT**
Accuracy	79.03%	80.23%	79.35%

6 Threats to Validity

The main external threat to the validity is the dataset we used. In our study, we utilize the widely used dataset collected by Sarwar et al. [2] and convert it to different data groups for binary classifications. But the labeling process of the data may not be accurate.

7 Conclusion

In this paper, we conduct an extensive study to evaluate the performance of multi-label commit message classification by applying p-tuning [10] on three pre-trained models such as BERT [4], RoBERTa [9] and DistilBERT [16]. In our study, we propose to convert the task of multi-label commit message classification into several binary classification tasks, based on the number of labels in the dataset. The results demonstrate that RoBERTa model outperforms other two models in terms of the widely used evaluation metrics (e.g., achieving 81.99% F1 score).

References

1. Meng, Na and Jiang, Zijian and Zhong, Hao. Classifying code commits with convolutional neural networks. 2021 International Joint Conference on Neural Networks (IJCNN). 2021
2. Sarwar, Muhammad Usman and Zafar, Sarim and Mkaouer, Mohamed Wiem and Walia, Gursimran Singh and Malik, Muhammad Zubair. Multi-label classification of commit messages using transfer learning. 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2020
3. Sanh, Victor and Debut, Lysandre and Chaumond, Julien and Wolf, Thomas. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. 2019
4. Devlin, Jacob. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
5. Mariano, Richard VR and dos Santos, Geanderson E and de Almeida, Markos V and Brandão, Wladmir C. Feature changes in source code for commit classification into maintenance activities. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA).2019
6. Brown, Tom B. Language models are few-shot learners. arXiv preprint arXiv:2005.14165. 2020
7. Luo, Xianchang and Xue, Yinxing and Xing, Zhenchang and Sun, Jiamou. Prcbert: Prompt learning for requirement classification using bert-based pretrained language models. Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, 2022
8. Schick, Timo and Schütze, Hinrich, Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676. 2020
9. Liu, Yinhan and Ott, Myle and Goyal, Naman and Du, Jingfei and Joshi, Mandar and Chen, Danqi and Levy, Omer and Lewis, Mike and Zettlemoyer, Luke and Stoyanov, Veselin. Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692, 2019
10. Liu, Xiao and Zheng, Yanan and Du, Zhengxiao and Ding, Ming and Qian, Yujie and Yang, Zhilin and Tang, Jie. GPT understands, too. AI Open, 2023
11. Wu, Bozhi and Liu, Shangqing and Feng, Ruitao and Xie, Xiaofei and Siow, Jingkai and Lin, Shang-Wei. Enhancing security patch identification by capturing structures in commits. IEEE Transactions on Dependable and Secure Computing.2022
12. dos Santos, Geanderson E and Figueiredo, Eduardo. Commit Classification using Natural Language Processing: Experiments over Labeled Datasets.CIbSE. 2020.
13. Ghadhab, Lobna and Jenhani, Ilyes and Mkaouer, Mohamed Wiem and Messaoud, Montassar Ben. Augmenting commit classification by using fine-grained source code changes and a pretrained deep neural language model.Information and Software Technology.2021
14. Heričko, Tjaša and Šumak, Boštjan and Karakatič, Sašo. Commit-Level Software Change Intent Classification Using a Pre-Trained Transformer-Based Code Model. Mathematics. 2024
15. Zeng, Qunhong and Zhang, Yuxia and Sun, Zeyu and Guo, Yujie and Liu, Hui. COLARE: Commit Classification via Fine-grained Context-aware Representation of Code Changes. 2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2024
16. Sanh, Victor and Debut, Lysandre and Chaumond, Julien and Wolf, Thomas. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108. 2019
17. Loshchilov, I. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101. 2017