Pull Request Generator

Solomatin Roman, Utkov Alexander, Vorkhlik Alexander, Shaforost Natalia

May 20, 2024

Abstract

This paper proposes a novel approach for generating pull request titles and descriptions from code diffs. Leveraging the capabilities of Large Language Models, we streamline the creation of informative and comprehensive pull request descriptions, enhancing the efficiency and quality of code reviews on platforms like GitHub.

Github repository

1 Introduction

In the era of digital transformation, where software development is fundamental for technological advancement, high-quality project documentation is crucial. Documentation serves as a tool for developers and users, providing insights into software functionality, usage, and updates. However, creating detailed and accurate documentation is often a time-consuming and labor-intensive task.

To address this issue, we explore the potential of AI-driven automation, specifically using modern natural language processing techniques. Large Language Models (LLMs) have shown remarkable capabilities in generating human-like text, making them ideal candidates for automating documentation tasks. This paper focuses on the application of LLMs to generate pull request (PR) descriptions from code diffs, therefore streamlining the code review process and improving documentation quality.

Recent studies indicate a growing trend among developers to utilize LLMs for PR title and description generation[19]. This approach not only saves time but also ensures consistency and clarity in the documentation. Our research aims to enhance this process by developing models that can generate comprehensive PR descriptions from code diffs, facilitating better project documentation in collaborative development environments.

2 Related Works

2.1 Pull Requests

In the realm of software development, the utilization of pull requests (PRs) is fundamental for facilitating collaborative work and reviewing code modifications. Recent endeavors have leveraged Large Language Models (LLMs) to automate the generation of clear PR descriptions, thereby enhancing developer productivity and optimizing the code review process[19].

The task of writing PR description often involves the creation of both a title and a detailed description. Some researchers focusing only on generating title from commits and issues[3, 10, 21]. Other uses sources such as commit messages and code comments for generating both title and description generation, but not using transformers[12]. Most recent work using encoder-decoder Transformer[16] architecture, but uses only commit messages for generation[5].

2.2 Code LLM

In recent years, several advanced code language models have been developed to enhance the understanding and generation of programming code. Notable among these are CodeT5[17], CodeT5+[18], DeepSeek-Coder[7], LLaMA[15, 14], and Phi[9, 1]. CodeT5 and its successor, CodeT5+, leverage the transformer-based T5 architecture, demonstrating exceptional versatility and performance across various code-related tasks such as code summarization, translation, and defect detection. CodeT5+ further improves upon its predecessor with enhanced pre-training and a larger model size, resulting in greater accuracy and efficiency. DeepSeek, on the other hand, is optimized for code search and retrieval, effectively understanding code context to generate meaningful summaries of code changes. LLaMA offers a scalable family of models pre-trained on a massive dataset, including code, making it highly adaptable for fine-tuning on specific tasks like code generation and summarization. Phi also utilizes state-of-the-art transformer architecture, excelling in precision and adaptability for various programming languages and styles. These models represent significant advancements in the field of code comprehension, facilitating the automatic generation of pull request descriptions from diffs, thereby improving the efficiency and quality of software development processes.

3 Proposed Approach

3.1 Model

The LLaMA-3 and Phi-3 models were selected as the base models for training due to their superior performance on HumanEval[3] and MBPP[2], as well as their ranking among the top two open models under 8 billion parameters on Chatbot Arena[4]. The Unsloth framework[1] was employed for training, enabling the use of LoRA[8] adapters with 4-bit quantized LLMs[6], which efficiently manage low memory requirements.

3.2 Dataset

For training and testing the models, we utilized merged pull requests from the Top100 Python repositories[2]. Specifically, we collected the title, description, and diffs for each pull request from these repositories[3].

Focusing on a single programming language enhances the model's ability to generate accurate descriptions. We included only merged pull requests to filter out suboptimal or random submissions and excluded repositories where the primary language is Chinese. From the dataset, we sampled 6,000 PRs with code diffs not exceeding 4,096 tokens. From this sample, we selected 1,000 PRs for validation and testing, ensuring an equal proportion from different repositories across all datasets[4].

3.3 Setup

3.3.1 Metrics

In our project, we used several established metrics to evaluate the performance of our models: ROUGE[11], ChrF++[13], and BERTScore[20]

3.4 Baselines

We compared our LLama-3 and Phi-3[1] fine-tuned on 4,000 PRs using Unsloth[5] with base models1, 3, 2.

[1] `https://github.com/unslothai/unsloth`

[2] `https://github.com/EvanLi/Github-Ranking/blob/933f2e85f64d30cf2a441fb5f43ebb41a9ec49bb/Top100/Python.md`

[3] `https://huggingface.co/datasets/Samoed/PRGen`

[4] `https://huggingface.co/datasets/Samoed/PRGenSelected`

[5] `https://github.com/unslothai/unsloth`

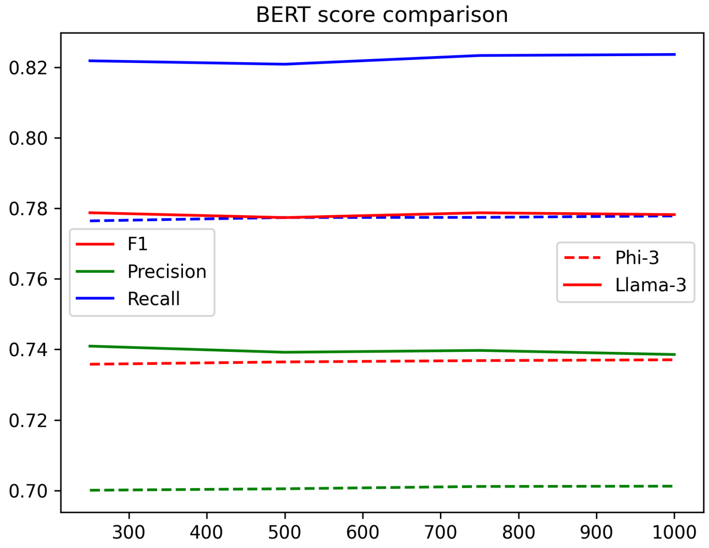

Figure 1: Comparison BertScore Llama 3 and Phi 3 during training

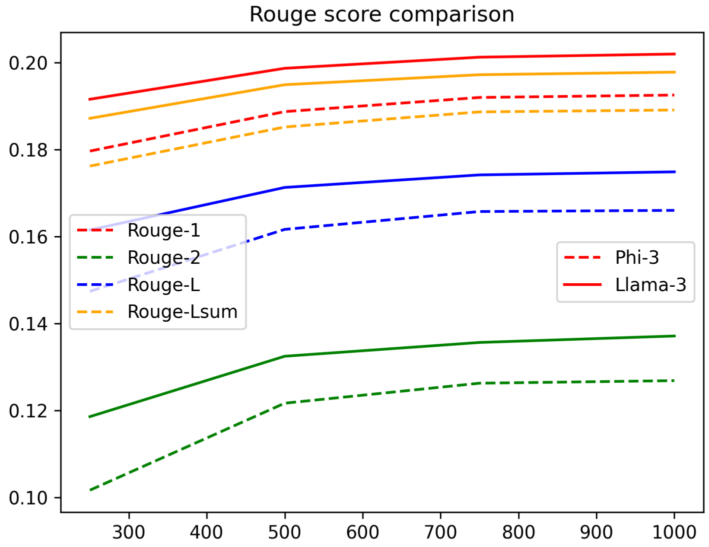

Figure 2: Comparison Rouge Llama 3 and Phi 3 during training

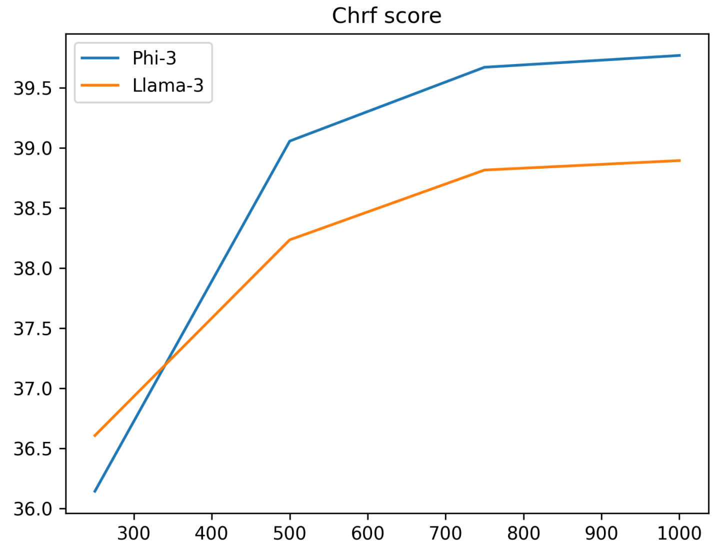

Figure 3: Comparison Chrf++ Llama 3 and Phi 3 during training

	rouge1	rouge2	rougeL	rougeLsum	CHRF++
LeadCM*	0.2019	0.0922	0.1859		
PRHAN*	0.2815	0.2011	0.3084		
Phi-3	0.1183	0.0272	0.0859	0.1026	9.6831
LLama-3	0.1520	0.0417	0.1216	0.1345	7.3922
LoRA-Phi-3	0.1275	0.0462	0.1151	0.1198	5.4918
LoRA-LLama-3	0.1549	0.0636	0.1460	0.1490	2.7478

Table 1: Description generation. *Results from [5]

After that we evaluate our models on evaluation dataset. Firstly we evaluate our model on description generation 1, 3. During evaluation it appeared that our models had issues generating PR descriptions, therefore we additionally add metrics for title generation only 2, 4. Results indicate that our fine-tuned models show superiority over the base models. We have found that our LLaMA model generated better texts comparing to Phi model.

Our generated results on eval dataset available here[6].

[6] `https://github.com/Samoed/PRGenerator/tree/main/notebooks/output`

	rouge1	rouge2	rougeL	rougeLsum	CHRF++
PRSummarizer*	0.3791	0.1799	0.3498		
Phi-3	0.0839	0.0260	0.0718	0.0719	12.0729
LLama-3	0.1905	0.0631	0.1686	0.1690	19.2157
Tuned Phi-3	0.2162	0.0848	0.2040	0.2041	15.8938
Tuned LLama-3	0.3376	0.1507	0.3216	0.3217	25.5881

Table 2: Title generation. *Results from [12]

Model Name	Precision	Recall	F1
Phi-3	0.8226	0.8011	0.8105
LLama-3	0.8516	0.8098	0.8292
LoRA-Phi-3	0.8534	0.7991	0.8244
LoRA-LLama-3	0.8713	0.8039	0.8353

Table 3: Bert Score on eval dataset for description generation

Model Name	Precision	Recall	F1
Phi-3	0.8136	0.8438	0.8273
LLama-3	0.8446	0.8648	0.8540
Tuned Phi-3	0.8562	0.8552	0.8550
Tuned LLama-3	0.8804	0.8755	0.8776

Table 4: Bert Score on eval dataset for title generation

4 Conclusions

In this paper, we looked over the history of research of pull request summarization and developed a model that could generate pull requests title and description based on commit history and code changes. From Github Top-100 python repositories we sampled 4,000 pull requests for training and 1,000 for validation and testing. Our findings showed that LLaMA model is superior in generating appropriate texts.

References

[1] Marah Abdin et al. *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone.* 2024. arXiv: `2404.14219 [cs.CL]`.

[2] Jacob Austin et al. *Program Synthesis with Large Language Models.* 2021. arXiv: `2108.07732 [cs.PL]`.

[3] Mark Chen et al. *Evaluating Large Language Models Trained on Code.* 2021. arXiv: `2107.03374 [cs.LG]`.

[4] Wei-Lin Chiang et al. *Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference.* 2024. arXiv: `2403.04132 [cs.AI]`.

[5] Sen Fang et al. "PRHAN: Automated Pull Request Description Generation Based on Hybrid Attention Network". In: *Journal of Systems and Software* 185 (2022), p. 111160. ISSN: 0164-1212. DOI: `https://doi.org/10.1016/j.jss.2021.111160`. URL: `https://www.sciencedirect.com/science/article/pii/S016412122100248X`.

[6] Elias Frantar et al. *GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers.* 2023. arXiv: `2210.17323 [cs.LG]`.

[7] Daya Guo et al. *DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence.* Jan. 26, 2024. DOI: `10.48550/arXiv.2401.14196`. arXiv: `2401.14196 [cs]`. URL: `http://arxiv.org/abs/2401.14196` (visited on 02/07/2024). preprint.

[8] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models.* Oct. 16, 2021. DOI: `10.48550/arXiv.2106.09685`. arXiv: `2106.09685 [cs]`. URL: `http://arxiv.org/abs/2106.09685` (visited on 11/22/2023). preprint.

[9] Alyssa Hughes. *Phi-2: The Surprising Power of Small Language Models.* en-US. Microsoft Research. Dec. 12, 2023. URL: `https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/` (visited on 01/01/2024).

[10] Ivana Clairine Irsan et al. *AutoPRTitle: A Tool for Automatic Pull Request Title Generation*. Aug. 5, 2022. DOI: 10.48550/arXiv.2206.11619. arXiv: 2206.11619 [cs]. URL: http://arxiv.org/abs/2206.11619 (visited on 04/23/2024). preprint.

[11] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[12] Zhongxin Liu et al. *Automatic Generation of Pull Request Descriptions*. Sept. 16, 2019. DOI: 10.48550/arXiv.1909.06987. arXiv: 1909.06987 [cs]. URL: http://arxiv.org/abs/1909.06987 (visited on 04/23/2024). preprint.

[13] Maja Popović. "chrF: character n-gram F-score for automatic MT evaluation". In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 392–395. DOI: 10.18653/v1/W15-3049. URL: https://aclanthology.org/W15-3049.

[14] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. July 19, 2023. DOI: 10.48550/arXiv.2307.09288. arXiv: 2307.09288 [cs]. URL: http://arxiv.org/abs/2307.09288 (visited on 11/22/2023). preprint.

[15] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. Feb. 27, 2023. DOI: 10.48550/arXiv.2302.13971. arXiv: 2302.13971 [cs]. URL: http://arxiv.org/abs/2302.13971 (visited on 10/11/2023). preprint.

[16] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[17] Yue Wang et al. *CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation*. Sept. 2, 2021. DOI: 10.48550/arXiv.2109.00859. arXiv: 2109.00859 [cs]. URL: http://arxiv.org/abs/2109.00859 (visited on 03/01/2024). preprint.

[18] Yue Wang et al. *CodeT5+: Open Code Large Language Models for Code Understanding and Generation*. May 20, 2023. DOI: 10.48550/arXiv.2305.07922. arXiv: 2305.07922 [cs]. URL: http://arxiv.org/abs/2305.07922 (visited on 03/01/2024). preprint.

[19] Tao Xiao et al. *Generative AI for Pull Request Descriptions: Adoption, Impact, and Developer Interventions*. Feb. 14, 2024. DOI: 10.1145/3643773. arXiv: 2402.08967 [cs]. URL: http://arxiv.org/abs/2402.08967 (visited on 03/01/2024). preprint.

[20] Tianyi Zhang et al. *BERTScore: Evaluating Text Generation with BERT*. 2020. arXiv: 1904.09675 [cs.CL].

[21] Ting Zhang et al. *Automatic Pull Request Title Generation*. June 30, 2022. DOI: 10.48550/arXiv.2206.10430. arXiv: 2206.10430 [cs]. URL: http://arxiv.org/abs/2206.10430 (visited on 04/23/2024). preprint.

A Prompt

```
### Instruction:
You a helpful code assistant that generates a text
    description of a pull request based on the DIFF of
    pull request.
Your task is to provide a concise summary of the
    changes. This summary will be used as description of
     pull request.
You should output only the description of this DIFF (
    description of pull request). You should not include
     any other text.
You think deeply about the changes and carefully
    analyze them.
Example:
### DIFF:
diff a/main.py b/main.py @@ -1,4 +1,4 @@
a = 1
b = 2
- c = 3
+ c = 4
print(c)

# Answer:
Change value of c from 3 to 4

### DIFF: {}
```

### Answer: {}

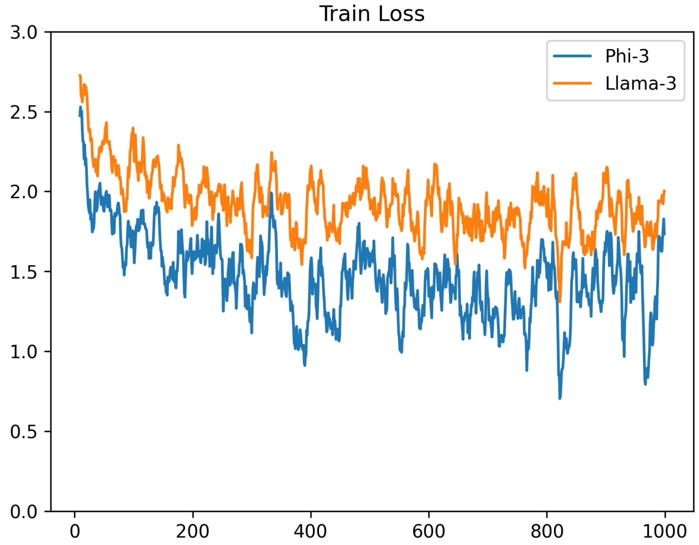

Figure 4: Loss

B Training

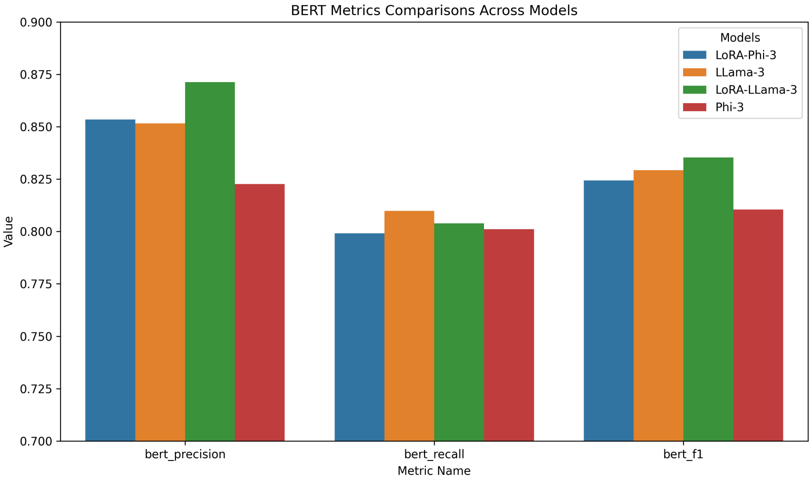

Figure 5: Bert score on eval dataset

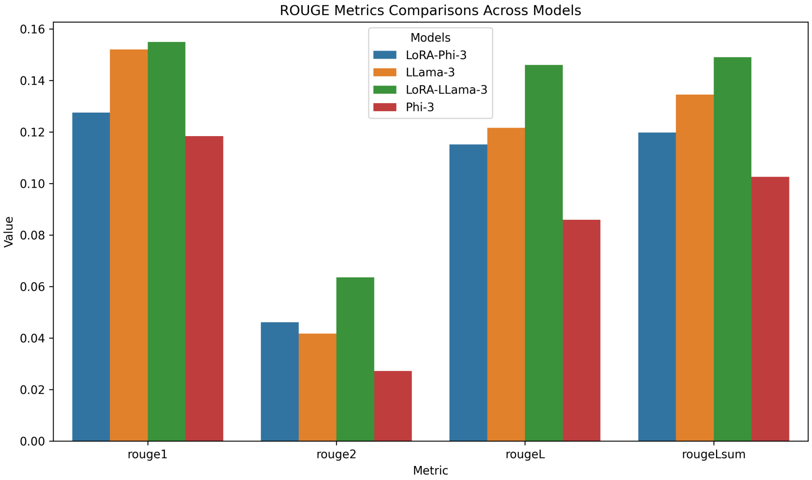

Figure 6: Rouge score on eval datset

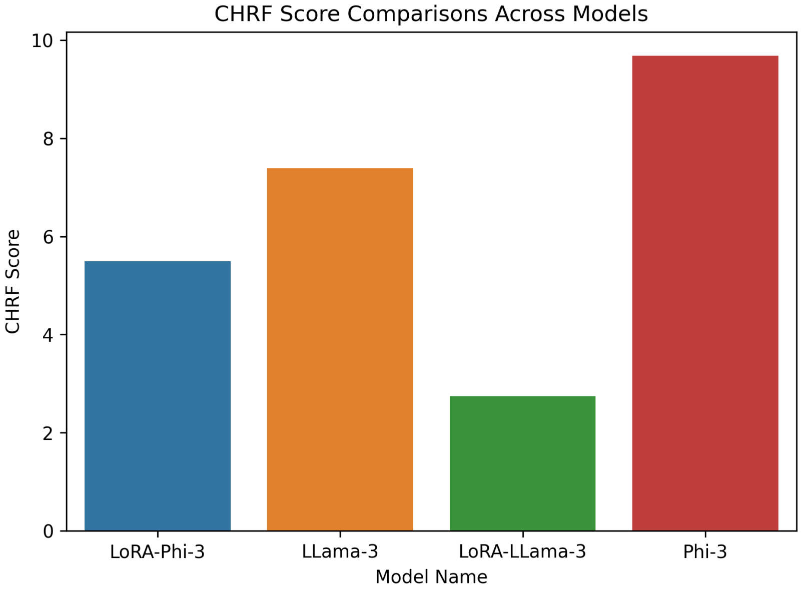

Figure 7: Chrf++ score on eval dataset