An Approach for Natural Language and Programming Language with Linux Kernel Patchwork Dataset

Jonghyeon Kim

Ajou University
tome01@ajou.ac.kr

Abstract

Many researchers and industries conducted a study for popular programming languages such as Python and JavaScripat. But, the relatively fewer focused programming language like C/C++ has lacks of study aspect of natural language processing. I found this skewed situation for many used PL has more dataset. To overcome this starvation of NL-PL pair dataset for low-level PL, we need to find more dataset for low-level PL, such as Linux Kernel Mailing List. By collecting dataset from the kernel patchwork mailing list, we could see more room for c languages such as code summarization and generation and sentimental analysis for source reviewers.

Introduction

Natural Language Processing (NLP) is a hot topic of the computer science academic, and fastly innovated and studied to various field. With the advanced neural network, much prior work investigated the relationship between programming language and natural language. As a result, code generation with open source questions and community answers like Stack overflow and code summarization.

However, Stack overflow users' Question and Answer dataset is skewed because popular Programming languages exist. Machine Learning variously uses Python researchers, Data scientists, and web programmers due to intuition and is more comprehensive than other languages. JavaScript is also popularly used by Web programmers. For these reasons, the Stack Overflow dataset has its information.

System software programmers like Linux kernel developers commit their code modification for Kernel source code, and others review the commit and give some opinion for a patch. Kernel maintainer decides whether affect this patch to mainline kernel tree or not. This flow of kernel patchwork is recorded to Linux Kernel Mailing List(LKML) because they used the mail to commit the kernel patch. With this data in LKML, we can find the positive and negative comments for patch commit and code summarization for the difference of code than before, and we also have a room for generating code-related optimization and fixation of kernel bugs.

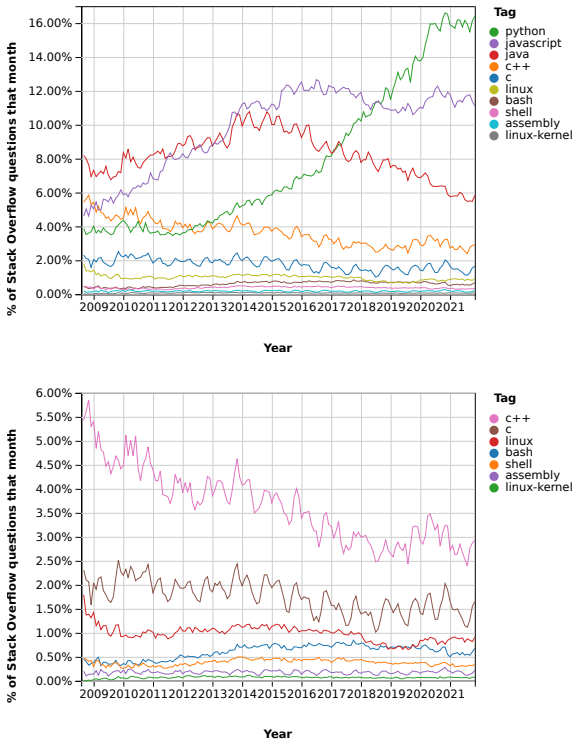

Figure 1: Stack Overflow trends of most popular languages for last decade(top) and the trends of system-related languages for stack overflow(bottom)

Motivation

With the existing dataset, natural language(NL) and programming language(PL) code snippet pair, most question and answer dataset for low-level PL like C/C++ and interpreter language related to computer systems like bash shell is not relatively focused on code generation and summarization tasks, which is because of meaning representation difficulty.

In prior works about NL to code generation and code summarization (Xu et al. 2020; Orlanski and Gittens 2021; Yin et al. 2018; Parvez et al. 2021; Feng et al. 2020). Their approaches are mainly based on many used PL such as Python,

Java, JavaScript, etc. It may be because those PL have numerical libraries and documentations. Besides, the number of users using high-level PL is also relatively more significant than users using low-level PL such as C/C++, Assembly code, and shell script languages such as sh, csh, and bash.

Figure 1(top) shows the most popular PL from Stack Overflow trends for the last decade using the questions tag. With the convenience and accessibility, Python has been the top popular language. After 2014, since Deep Neural Network(DNN) a framework like TensorFlow and PyTorch are developed and widely used, Python question ratio of SO significantly increased.

Figure 1(bottom) describes the usage of low-level PL and shell script languages. We can realize questions for those PL are not frequently asked by users due to a relative lack of users. But we know that a QA dataset for low-level PL is needed for many operating systems and computer architecture researchers and engineers because emerging hardware requires proper software support.

To expand the dataset for such non-focused PL, we need to collect from SO QA NL-PL code pair and the various opensource communities for system software languages such as Linux Kernel Mailing List(LKML) and Ask Ubuntu for shell. There is the LKML archive dataset from Keggle (Miasoedov 2017), but there are no codes and discussions for this dataset.

Using the LKML dataset, I anticipate that the sentimental analysis for committing with review messages from kernel maintainer is possible to exploit that dataset. another thing what I proposed method for that dataset is code summarization for code patches. Most Linux kernel patch has code difference compared to the previous version of the kernel and commit message describing what they add new features or remove unnecessary code for readability or fix kernel panic bugs. We can exploit those information from kernel patch mail to summarize code or describing why they optimize code for their goals.

Related Work

CoNaLa (Yin et al. 2018) dataset extracted from STACK OVERFLOW (SO) with three specific elements, *Intent*, *Context*, and *Snippet*. Intent is a description in English of what the question wants to do. Context is a piece of code that does not implement the intent, but is necessary setup. Snippet is a piece of code that actually implements the Intent. Then, they learned learned from a small number of annotated example, which made the code/natural language pair dataset from the SO.

In (Orlanski and Gittens 2021), they expanded CoNaLa dataset by adding the textual question bodies from StackExchange API and comibined simple BART (Lewis et al. 2019) model to generate answering code for quesiton. By adopting BART, they improved model performance (BLEU score) as mush as state-of-the-art.

In (Xu and Zhou 2018), they presented Multi-level dataset of LKML project on the Linux kernel patchwork. They deal with the dataset divided 3-level. Level 0 data contain raw data for LKML files. Level 1 data is stored by MySQL database from level 0 data. Level 2 data is more categorized explicitly from level 1 data, reducing researcher effort in collecting, cleaning, and processing patch data.

Problem Definition

The main approaches to exploit LKML patch dataset are following:

- Combine curretly exisintg LKML dataset (Miasoedov 2017; Xu and Zhou 2018)
- Sentimental analysis for maintainer review of each commit
- Code generation from patch of fixing kernel bug and optimizing kernel code with thier mentions
- Code summarization from new patch description with changelogs and code differfence

Conclusion

We surveyed the dataset for a question and answer to generate code as an answer, which mainly focused on popular a programming language such as Python and JavaScript. Although computer system-related PL lacks users compared to other popular PL, there is also helpful to users making dataset. By using an existing dataset for the Linux Kernel Mailing List dataset, we have more possibility to data processing with the neural network, code generation and summarization, and sentimental analysis for kernel developers.

References

Feng, Z.; Guo, D.; Tang, D.; Duan, N.; Feng, X.; Gong, M.; Shou, L.; Qin, B.; Liu, T.; Jiang, D.; and Zhou, M. 2020. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. arXiv:2002.08155.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461*.

Miasoedov, A. 2017. Linux Kernel Mailing List archive. https://www.kaggle.com/msoedov/linux-kernel-mailing-list-archive.

Orlanski, G.; and Gittens, A. 2021. Reading StackOverflow Encourages Cheating: Adding Question Text Improves Extractive Code Generation. arXiv:2106.04447.

Parvez, M. R.; Ahmad, W. U.; Chakraborty, S.; Ray, B.; and Chang, K.-W. 2021. Retrieval Augmented Code Generation and Summarization. In *EMNLP-Findings*.

Xu, F. F.; Jiang, Z.; Yin, P.; and Neubig, G. 2020. Incorporating External Knowledge through Pre-training for Natural Language to Code Generation. In *Annual Conference of the Association for Computational Linguistics*.

Xu, Y.; and Zhou, M. 2018. A Multi-Level Dataset of Linux Kernel Patchwork. In *Proceedings of the 15th International Conference on Mining Software Repositories*, MSR '18, 54–57. New York, NY, USA: Association for Computing Machinery. ISBN 9781450357166.

Yin, P.; Deng, B.; Chen, E.; Vasilescu, B.; and Neubig, G. 2018. Learning to Mine Aligned Code and Natural Language Pairs from Stack Overflow. In *International Conference on Mining Software Repositories*, MSR, 476–486. ACM.