BFR: Bug Fixing Patches Ranking Model in Linux Kernel

ABSTRACT

ACM Reference Format:
. 2017. BFR: Bug Fixing Patches Ranking Model in Linux Kernel. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 2 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

2 RELATED WORK

3 CONVOLUTIONAL NEURAL NETWORKS FOR SENTENCE CLASSIFICATION

4 APPROACH

4.1 Formalization

Let S be a set of stable bug fixing patches and P a set of all bug fixing patches in Linux kernel. We call $U \in (S \cap P)$ is a set of unknown bug fixing. In our problem, we try to provide a bug fixing patches ranking $\subset P^2$ for a developer, where each bug fixing patch has to meet the following properties [2]:

- *Totality:* $\forall i, j \in P : i \neq j \Rightarrow (i > j) \vee (j > i)$
- *Antisymmetry:* $\forall i, j \in P : (i > j) \wedge (j > i) \Rightarrow i = j$
- *Transitivity:* $\forall i, j, k \in P : (i > j) \vee (j > k) \Rightarrow i > k$

4.2 Bug Fixing Patch Ranking

The architecture of bug fixing patch ranking is represented in Fig. 1. Our model based on convolutional neural network [1] to map bug fixing patches to vectors, which can be used to compute their ranking score. In the following, we describe how the model computes the ranking score given by a pair of bug fixing patches. We briefly explain each component (i.e., hidden layer, logistic function, etc.) of our model.

4.3 Learning Relationship Between Text and Code in Single Patch

In Linux kernel, each patch contains both a textual commit message and commit code. The commit message, which can help a developer to speed up the reviewing process, is a short description of source code change. The commit code is the changes that are applied on the buggy file.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

4.4 Hidden Layer

4.5 Feature Representation

4.6 Output Layer

4.7 Training

Our model is trained by minimizing the cross-entropy lost function:

$$\begin{aligned}\mathcal{L} &= -\log \prod_{i \in S, j \in U}^{\mathrm{N}} p(y_{ij} | p_i, p_j) + \lambda \|\theta\|_2^2 \\ &= -\sum_{i \in S, j \in U}^{\mathrm{N}} \log \sigma(\mathbf{a}_{ij}) + \lambda \|\theta\|_2^2\end{aligned} \tag{1}$$

where S and U are the stable and unknown bug fixing patches datasets respectively, $y_{ij} = 1$ since p_i has higher rank compared to p_j, N is the total number pairs of bug fixing patches in S and U, $\sigma(.)$ represents the logistic function to identify the rank order of patches pair. $\mathbf{a}_{ij}$ is the output of our ranking model and can be decomposed as:

$$\mathbf{a}_{ij} = \mathbf{a}_i - \mathbf{a}_j \tag{2}$$

to satisfy the properties mentioned in Sec. 4.1.

4.8 Regularization

REFERENCES

[1] Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.

[2] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence.* AUAI Press, 452–461.

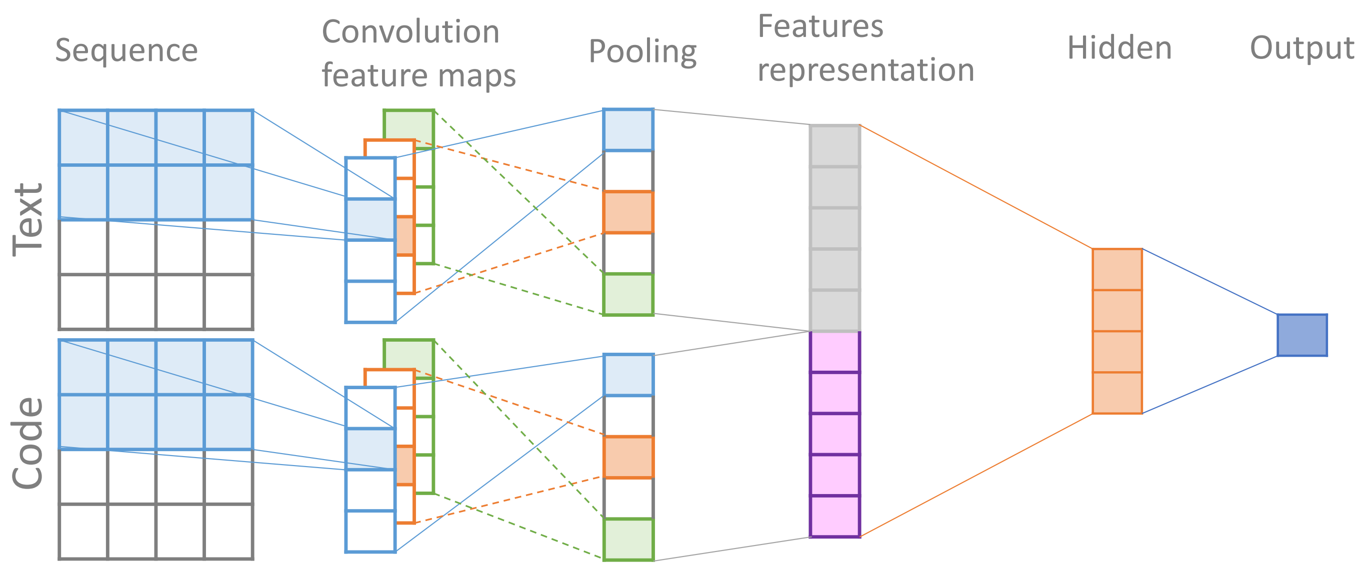

Figure 1: An architecture of bug fixing patches ranking model in Linux kernel.