Data Augmentation for Boosting Source Code Learning

董, 沢銘

https://hdl.handle.net/2324/7182485

出版情報：Kyushu University, 2023, 博士（工学）, 課程博士
バージョン：
権利関係：

（様式 2）

氏　　名　：董　沢銘

論 文 名　：Data Augmentation for Boosting Source Code Learning
（ソースコード学習を促進するためのデータ拡張）

区　　分　：甲

論文内容の要旨

With the rise of deep code models, we are witnessing the success of code intelligence. However, current research mainly focuses on developing new code models or fine-tuning large pre-trained models for code-related tasks. One significant issue that often remains overlooked is the preparation of high-quality data to boost the training of these code models. In contrast, the importance of data quality and diversity via data augmentation techniques is well-established in fields such as computer vision and natural language processing. Surprisingly, this aspect has not been extensively explored in source code learning, and most existing practices of data augmentation in source code learning are limited to simple syntax-preserved methods such as code refactoring and are not sufficiently effective.

To address these challenges and contribute to the advancement of data augmentation in source code learning, this thesis primarily focuses on enriching effective and reliable code data augmentation methods. These include syntax-preserved methods and syntax-broken methods, and the work is divided into three phases:

1. Essentially, source code is often represented sequentially as text data. Inspired by this analogy, we take an early step to investigate whether data augmentation methods originally for text data are effective for source code learning. Our focus centered on understanding tasks, and we conducted a comprehensive empirical study spanning four critical code-related problems and four deep neural network (DNN) architectures. Our findings identify that the data augmentation methods can produce more accurate and robust models for source code learning and show that the augmented data are still useful even if they slightly break the syntax of the source code.

2. Sequential representation ignores the semantic information of the source code. To overcome this limitation, the graph-structural representation of the source code is proposed. In light of the graph nature of the source code, we propose to apply the data augmentation methods used for graph-structured data in graph learning to the tasks of source code learning. We conduct a comprehensive empirical study to evaluate whether such new ways of data augmentation are more effective than the existing simple code refactoring methods in terms of producing more accurate and robust models. Specifically, we evaluate four critical software engineering tasks and seven neural network architectures to assess the effectiveness of five data augmentation methods. Our findings identify that linear interpolation can significantly improve both the accuracy and robustness of the trained models for source code learning.

3. We introduce a data augmentation approach, MixCode, that draws inspiration from Mixup in computer vision. MixCode aims to effectively supplement valid training data without manually collecting or labeling new code. Specifically, we first employ code refactoring methods to create transformed code instances that maintain consistent labels with the original data. Subsequently, we adapt the Mixup to combine the original code with these transformed code samples, thereby augmenting the training data. Our evaluation of this approach includes two programming languages, two understanding tasks, four benchmark datasets, and seven different model architectures. Our

Experimental results demonstrate that MixCode outperforms baseline data augmentation approaches in terms of accuracy and robustness.

In summary, this thesis fills the gaps in effective and reliable data augmentation for source code learning by (1) introducing text-oriented data augmentation for source code learning, (2) providing graph-oriented data augmentation for source code learning, and (3) developing MixCode, the first online code data augmentation method.