Overview

We have code PLMs and abundant code data on GitHub, but not much for specific downstream tasks like code translation and summarization. Hence we explore data augmentation:

1. monolingual - abundant resources.
2. multilingual - similarity between programming languages
3. numerical - code correctness.

Language Utilization

Programming languages share higher similarities, such as numbers, syntax tokens, etc. We study the use of other languages in two ways: autoencoding the target language for translation, and multilingual training for code summarization.

Data synthesis

We apply back-translation to obtain pseudo-parallel data from monolingual code for translation. For code summarization, we first reverse its data to train a multilingual text-to-code generator, then generate code in arbitrary programming languages to pair with genuine text summaries.

Numeric Awareness

We propose a novel numeric encoding method to let numbers pass through the network as illustrated in Figure 1. Apart from this, for code translation, we swap numbers on both input and output ends consistently to create extra data.

Task, Data, and Metrics

We present the results of C#-to-Java translation and summarization using CodeBERT. Please check out our paper for training configurations, more results on other PLMs, as well as our explorations on the code synthesis task.

Code outputs are evaluated by exact line match, BLEU and CodeBLEU, whereas text summaries are evaluated by BLEU.

Acknowledgements

Pinzhen Chen is funded by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070350 and from UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee (grant number 10052546 - High Performance Language Technologies). This presentation is partially funded by Institute for Language, Cognition and Computation, School of Informatics, University of Edinburgh. We thank the MindSpore team for providing technical support.