CODE-TO-CODE TRANSLATION: Java → C# ELIMINATE LEGACY CODE

Frederick Behringer, Robin Dederichs, Marcel Struck

1: Motivation

Code-to-code translation is an often overlooked application of natural language processing (NLP) with substantial implications for software development and maintenance. As programming languages evolve and new languages emerge, the need for translating legacy code into modern languages becomes increasingly crucial. This process helps in preserving the functionality of existing software while leveraging the advantages of newer languages.

Similarly, developers often need to port software from one language to another to improve performance, enhance readability, or integrate with other systems. Manual translation of code is a time-consuming and error-prone task that requires deep understanding of both source and target languages.

By automating code translation, we aim to reduce the manual effort required, minimize translation errors, and facilitate seamless integration between different programming environments. Our focus is on studying the capabilities of NLP models to translate code accurately, ensuring that both syntax and semantics are correctly preserved after translation. This will ultimately boost productivity and code quality in the software development lifecycle.

2: Approach

To address the challenges imposed by code translation, we utilize the CodeXGLUE code-to-code translation dataset [1], which provides equivalent code snippets in C# and Java. The dataset is already split into a training (10,300 samples), validation (500 samples), and test set (1,000 samples), to enhance evaluation over multiple approaches.

The model we choose to perform code-to-code translation is CodeT5+ [2], since it is specifically optimised on code tasks. At the time of writing, CodeT5+ has not been evaluated for this task and has not seen the dataset. We employ three versions of the CodeT5+ 220M model to solve the task. First, we evaluate the performance of the pre-trained model. Second, we fine-tune the model on our dataset. Third, we train it from scratch, leaving out any pre-training. All three versions are compared to find the best suiting apporach. Figure 2 visualizes our proceeding.

To evaluate the performance of the models we use the BLEU (bilingual evaluation understudy) and CodeBLEU [3] metrics. CodeBLEU is specifically designed for code translation and assesses syntactic, semantic, and structural accuracy (Figure 1). BLEU provides a broader comparison to other NLP models, ensuring our results are comparable across different benchmarks.

In addition to our modified models, we benchmark other existing models in the field. Shuai Lu et al.[1] provide values for multiple models from 2021. To include a more recent candidate, we also evaluate the general-purpose Llama 3 with zero-shot prompting.

By systematically evaluating our models using these metrics, we aim to demonstrate the effectiveness of pre-training and provide insights into the capabilities and limitations of NLP models in the context of code-to-code translation.

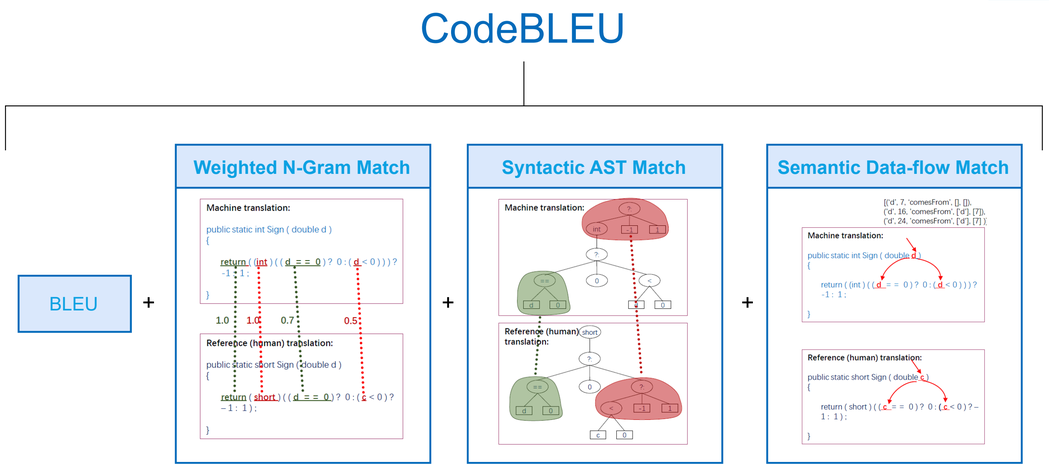

Figure 1: A visualisation of the CodeBLEU metric [3]

3: Evaluation

Table 1 presents the evaluation results of our fine-tuned model in comparison to various other models on the same code-to-code task. Our fine-tuned CodeT5+ model outperforms the referenced models from [1] as well as CodeT5+ without pre-training and zero-shot prompting on the general-purpose LLM Llama 3 8B. Especially the BLEU score is significantly higher and reaches a near-optimal performance of 98.44 after ten epochs. Also interesting to note is that the performance increases drastically, when compared to the unoptimized CodeT5+ model.

Method	BLEU	CodeBleu
Naive Copy [1]	18.54	-
PBSMT [1]	43.53	42.71
Transformer [1]	55.84	63.74
Roberta (code) [1]	77.46	83.07
CodeBERT [1]	79.92	85.10
LLaMA3 8B (zero-shot-prompting)	49.07	35.79
CodeT5+ 220M	0.0	13.31
CodeT5+ 220M self trained	0.0	20.49
CodeT5+ 220M fine-tuned	**98.44**	**87.55**

Table 1: Evaluation of Java → C# translation

4: Takeaways

The results shows the effectiveness of fine tuning a pre-trained model and explains the wide spread distribution of this attempt in the industry. Without fine-tuning the model performs poorly on the given task of translating Java code to C#, which is represented by a BLEU-score of 0.0. The fine-tuned model shows a surprisingly high BLEU and CodeBLEU score. Notably, the results of the fine-tuned model were achieved using a relatively small version of CodeT5+ with 220M parameters, whereas most of the compared models are significantly larger. Thus, it is to be expected that fine-tuning on a larger model, with 6 or 16 Billion parameters, would offer even better results.

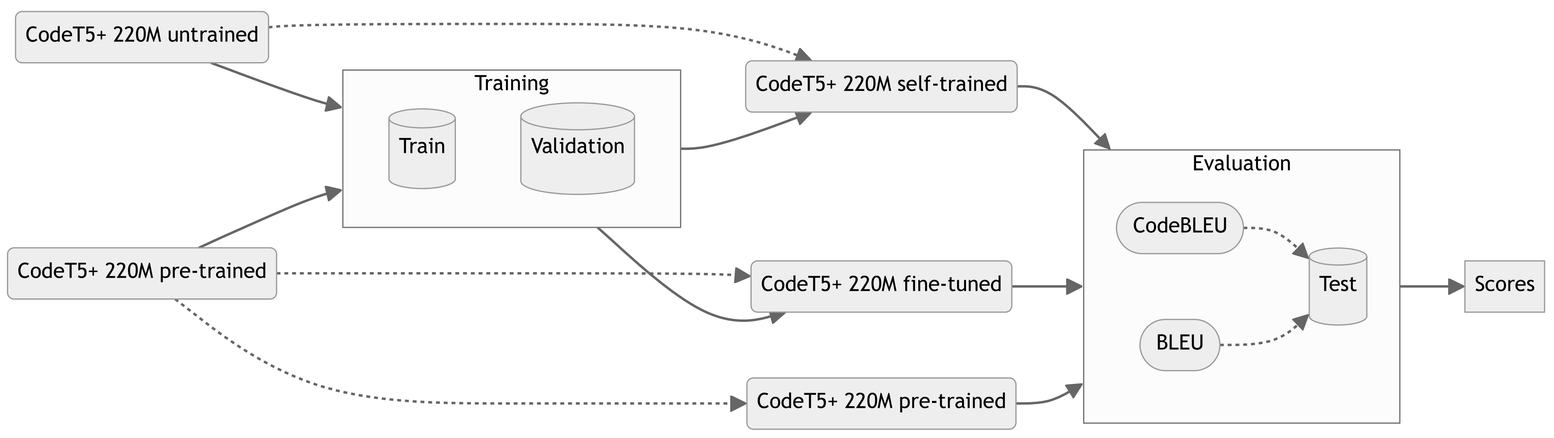

Figure 2: An Overview of our approach

References

[1] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin B. Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation. *CoRR*, abs/2102.04664, 2021.

[2] Yue Wang, Hung Le, Akhilesh Deepak Gotmare, Nghi D.Q. Bui, Junnan Li, and Steven C. H. Hoi. Codet5+: Open code large language models for code understanding and generation. *arXiv preprint*, 2023.

[3] Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, Neel Sundaresan, Ming Zhou, Ambrosio Blanco, and Shuai Ma. Codebleu: a method for automatic evaluation of code synthesis. *CoRR*, abs/2009.10297, 2020.

[4] Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *EMNLP*, 2021.

Acknowledgement

We would like to express our gratitude to our supervisors, Prof. Dr. Jörn Hees and M.Sc.Tim Metzler, for their motivation and guidance.

Contact

Frederick Behringer, Robin Dederichs, Marcel Struck
Hochschule Bonn-Rhein-Sieg
Email: [name].[lastname]@smail.inf.h-brs.de

b-it Bonn-Aachen International Center for Information Technology

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences