Global Conference on Sustainable and Futuristic Technologies 2023
(GConSFT-2023)

Analysing Code-BERT and its efficacy for the code-comment relationship prediction task (Paper ID-165)

Presented by :-

Mr. Akshay Dongare

BE Student, Computer Department, Marathwada Mitra Mandal's College of Engineering

Guided by :-

Dr. K. S. Thakre

Professor and Head of Department, Computer Department, Marathwada Mitra Mandal's College of Engineering

TABLE OF CONTENTS

INTRODUCTION

- How can machine learning can assist software developers?
- Big Data versus Big Code (large repositories of programs)
- Google and Alphabet Kaggle Competitions
- Why python notebooks?

Motivation

An understanding of the relationships between code and markdown could lend to fresh improvements across many aspects of AI-assisted development

Examples:

- Construction of better data filtering and preprocessing pipelines for model training
- Automatic assessments of a notebook's readability

Objectives

- Study existing models in this domain
- To build an AI Model capable of predicting the correct relationship between comments and code
- Evaluation of proposed prediction model using the Kendall Tau correlation

Methodology

- Reconstructed the order of markdown cells in a given notebook based on the order of the code cells, demonstrating comprehension of which natural language references which code

- A dataset of approximately 160,000 public ipython notebooks from Kaggle is currently used.

Literature Survey

REFERENCE	AUTHORS	TECHNOLOGY USED	KEYWORDS
[2]	Feng, Zhang Yin and Guo, Daya and Tang, Duyu and Duan	CodeBERT: A Pre-Trained Model for Programming and Natural Languages	Transformers, NLP, Language Models, BERT, GraphCodeBERT
[3]	Phan, Long and Tran, Hieu and Le, Daniel and Nguyen, Hieu and Anibal, James and Peltekian, Alec and Ye, Yanfang	ConTexT: Multi-task Learning with Code-Text Transformer	MTL (Multi-task Learning), BERT, GPT
[10]	Shoeybi, Mohammad and Patwary, Mostofa and Puri, Raul and LeGresley	Megatron-LM: Training Multi-Billion Parameter Language Models	BLOOM, GPT3, LM, STL, General AI
[9]	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	Encoder, Decoder, Big Code, NLP, pre-trained

CodeBERT: A Pre-Trained Model for Programming and Natural Languages

CodeBERT learns general-purpose representations that support downstream NL-PL applications such as natural language codesearch, code documentation generation, etc.

Advantages:

1. Transformer-based neural architecture
2. Hybrid Training objective (replaced token detection)
3. Bimodal Data (NL-PL pairs)

Disadvantages:

1. Context Fragmentation
2. Can only handle fixed-length text strings

CoTexT: Multi-task Learning with Code-Text Transformer

CoTexT is a pre-trained, transformer-based encoder-decoder model that learns the representative context between natural language (NL) and programming language (PL).

Advantages:

1. CoTexT supports downstream NL-PL tasks such as code summarizing/documentation, code generation, defect detection, and code debugging.

Disadvantages:

1. Tasks can compete with each other in order to achieve a better learning representation
2. More complex as a result of multiple summed losses thereby making the optimization difficult.

CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation

CodeT5 is a unified pre-trained encoder-decoder Transformer model that better leverages the code semantics conveyed from the developer-assigned identifiers

ADVANTAGES:

1. Multi-task learning (understanding and generation)
2. Identifier-aware pre-training feature
3. Consists of Encoder as well as Decoder

DISADVANTAGES:

1. Automation Bias
2. Security Implications

Evaluation Metric

Kendall’s Tau Correlation

- The Kendall tau correlation, which compares the anticipated and ground truth cell ordering over the whole set of test set notebooks, is used to assess the predictions.
- Determine S, the quantity of neighboring entry swaps required to convert the predicted cell order into the ground truth cell order.
- A notebook with n cells will require, at most, 1/2 n (n − 1) swaps to sort a projected order.
- We total the number of swaps from your anticipated cell order for every notebook in the test set, and we do the same for the worst-case number of swaps.

$$K = 1 - 4\frac{\sum_i S_i}{\sum_i n_i(n_i - 1)}$$

Result

- “id” refers to the unique id of the python notebook
- cell order specifies the predicted order of code-comment cells

id	cell_order
0009d135ece78d	0a226b6a ddfd239c 8cb8d28a c6cd22db 1372ae9b e25aa9bd 90ed07ab ba55e576 7f388a41 f9893819 2843a25a 39e937ec 06dbf8cf
0010483c12ba9b	7f270e34 54c7cab3 fe66203e 7844d5f8 5ce8863c 4a0777c4 4703bb6d 4a32c095 865ad516 02a0be6d
0010a919d60e4f	23607d04 b7578789 aafc3d23 bbff12d4 80e077ec 584f6568 89b1fdd2 b190ebb4 8ce62db4 ed415c3c d3f5c397 18ce8cc0 35cd0771 322850af 7f53de45 c069ed33 50bc28b3 5115ebe5 868c4eae 5e8c5e7e 1d4dbeae 4ae17669 3f4a105f 80433cf3 bac960d3 a4875f3f bd8fbd76 8a0842b8 0e2529e8 1345b8b2 ea06b4d0 52fe98c4 cdae286f 03cb1feb 724d27d3 4907b9ef 44eb815a d65238ba 3bff2378 7d157458 8679f842 641e45c1 83514fa3 7f6a2fa8 f9e38e5a b78215d1 e52e4a9e 982d964e 9f5d983e 22776759 ef01da10 e0bf4b8b 7317e652 5793f12e 3741e756 bc8eaa53 f7f2ce31 0115f7f5 21b6fb8f 177f908c 4356ab34 d2f722a5
0028856e09c5b7	eb293dfc 012c9d02 d22526d1 3ae7ece3

Conclusion

- We have developed a model capable of understanding and comprehending natural language as well as code.
- This model can predict the natural language comments for each code block with 92% accuracy on the Kendall Tau Correlation metric

Future scope

- Multi-task learning capabilities to our model for other software engineering tasks
- Pruning techniques to reduce the model size
- Explore non-machine-learning based approaches

References

- [1] Allamanis, Miltiadis and Barr, Earl T and Devanbu, Premkumar and Sutton, Charles, “A survey of machine learning for big code and naturalness”, ACM Computing Surveys (CSUR), Vol. 51, 5 May 2018.
- [2] Feng, Zhangyin and Guo, Daya and Tang, Duyu and Duan, Nan and Feng, Xiaocheng and Gong, Ming and Shou, Linjun and Qin, Bing and Liu, Ting and Jiang, Daxin and Zhou, Ming, CodeBERT: A Pre-Trained Model for Programming and Natural Languages, arXiv, 18 Sep 2020
- [3] Phan, Long and Tran, Hieu and Le, Daniel and Nguyen, Hieu and Anibal, James and Peltekian, Alec and Ye, Yanfang, CoTexT: Multi-task Learning with Code-Text Transformer, arXiv, 21 Jun 2021
- [4] Ahmed, Toufique and Devanbu, Premkumar, Learning code summarization from a small and local dataset, arXiv,2 Jun 2022
- [5] Garg, Spandan and Moghaddam, Roshanak Zilouchian and Clement, Colin B. and Sundaresan, Neel and Wu, Chen, DeepPERF: A Deep Learning-Based Approach For Improving Software Performance, arXiv, 27 Jun 2022
- [6] Xu, Frank F. and Alon, Uri and Neubig, Graham and Hellendoorn, Vincent J., A Systematic Evaluation of Large Language Models of Code, arXiv , 4 May 2022

References

- [7] Daya Guo1∗ , Shuo Ren2∗ , Shuai Lu3∗ , Zhangyin Feng4∗ , Duyu Tang5 , Shujie Liu5 , Long Zhou5 , Nan Duan5 , Alexey Svyatkovskiy6 , Shengyu Fu6 , Michele Tufano6 , Shao Kun Deng6 , Colin Clement6 , Dawn Drain6 , Neel Sundaresan6 , Jian Yin1 , Daxin Jiang7 , and Ming Zhou5 1School of Computer Science and Engineering, Sun Yat-sen University. 2Beihang University, 3Peking University, 4Harbin Institute of Technology, 5Microsoft Research Asia, 6Microsoft Devdiv, 7Microsoft STCA , GRAPHCODEBERT: PRE-TRAINING CODE REPRESENTATIONS WITH DATA FLOW, ICLR, 2021
- [8] Peng, Dinglan and Zheng, Shuxin and Li, Yatao and Ke, Guolin and He, Di and Liu, Tie-Yan, How could Neural Networks understand Programs?, arXiv, 31 May 2021
- [9] 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, arXiv, 2 Sep 2021
- [10] Shoeybi, Mohammad and Patwary, Mostofa and Puri, Raul and LeGresley, Patrick and Casper, Jared and Catanzaro, Bryan, Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism, arXiv, 13 Mar 2020

Thank You