TCEM: An exhaustive text-to-code evaluation metric covering edge cases for the code snippets

Wajahat Mirza[1], Jia Bokang[2], Domnica Dzitac[3]

New York University

(mwm356[1], bj798[2], did233[3])@nyu.edu

1. Introduction

Text to code is a useful tool for developers who often forget syntactic implementation of simple code snippets. A recently published dataset CoNaLa [1] contains over 600k automatically labeled and over 2,379 manually labeled and annotated train samples [2]. The current baseline for this task has been established with a traditional sequence-to-sequence, seq2seq model [3]. With the advent of modern transformer models, it is possible to improve on this baseline and demonstrate to the community the performance of the newer models, such as GPT-2 and GPT-3, on the task of text to code [4]. We also hope to investigate tweaks that can improve these transformer models on the task.

Furthurmore, the CoNaLa competition [5] utilizes a BLEU score [6] for evaluating the quality of code snippets. While BLEU is very efficient in evaluating large corpora, "it is unreliable at the sentence or sub-sentence levels, and with a single reference" [7]. Thus, *minutus* changes in model could result in a big difference in the veracity and evaluation of the code snippet. We also hope to improve this evaluation method.

Terms: OpenAI GPT-3, text-to-code, transformers, unit tests

2. Feasibility

We plan to train the model on the GPT-3 network through the use of OpenAI's APIs [8]. In addition, we will focus our efforts on investigating prompt design and parameter tweaking for the task of text to code. We intend to explore techniques for calibrating few-shot performance [9] on our task, which could boost our performance from a base GPT-3 model.

We plan to develop a new evaluation metric of unit test cases [10] for the CoNaLa dataset which will be evaluated against the BLEU model. These unit tests cases will c0ver edge cases of each training and test examples from the 2000 human labeled instances. We roughly estimate that each data point example from CoNaLa dataset will require on average 3 unit tests, therefore our evaluation metric will contain roughly 9000 unit tests.

3. Data and tools

3.1. Available

- CoNaLa dataset that is publicly available.
- Unit test module which is publicly available.
- HPC access that is provided through NYU Greene.

3.2. Required

- For model training, we require GPT-3.
- Currently, we have access to OpenAI tokens but our license expires in 2 weeks from today's date.
  - We may require funds and access to GPT-3 from OpenAI until May. The cost is approximately $0.06 per 1000 tokens.

4. Collaboration statement

All collaborators are working equally in developing this project. The brainstorming and development of the plan has been done in collaboration with researcher Tal Schuster and prof. Sam Bowman. The research and literature review will be conducted as a collaborative work, split equally between all members. While Bokang will be in charge of training the model with GPT-3, Mirza and Domnica will work on improving the performance of the model. Mirza and Domnica will also start building the theoretical bases for the new evaluation metric, TCEM. The whole team will contribute towards writing unit test cases for our TCEM metric.

5. Literature Review References

While literature review will be expanded as we progress, as of now, we have read or intend to read the following papers to enhance our knowledge for the project.

- Learning to Mine Aligned Code and Natural Language Pairs from Stack Overflow [1]
- Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task [11]
- Big Code != Big Vocabulary: Open-Vocabulary Models for Source Code [12]
- Code Completion with Statistical Language Models [13]
- BLEU Deconstructed: Designing a Better MT Evaluation Metric [7]
- Deep API Learning [14]
- How to evaluate machine translation: A review of automated and human metrics [15]
- Making Pre-trained Language Models Better Few-shot Learners [16]

6. Acknowledgements

We would like to extend our gratitude and appreciation towards Tal Schuster (CSAIL, MIT) and prof., Sam Bowman (CDS, NYU) for their time and guidance throughout the process.

7. References

[1] P. Yin, B. Deng, E. Chen, B. Vasilescu, and G. Neubig, "Learning to Mine Aligned Code and Natural Language Pairs from Stack Overflow," *arXiv e-prints*, p. arXiv:1805.08949, May 2018.

[2] "Conala: The code/natural language challenge," https://conala-corpus.github.io/, 2018.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *arXiv e-prints*, p. arXiv:1409.3215, Sep. 2014.

[4] "Next chapter in artificial writing," *Nature Machine Intelligence*, vol. 2, no. 419, 2020.

[5] "Codalab - competition," competitions.codalab.org/competitions/19175, 2018.

[6] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://www.aclweb.org/anthology/P02-1040

[7] X. Song, T. Cohn, and L. Specia, "Bleu deconstructed: Designing a better mt evaluation metric," *International Journal of Computational Linguistics and Applications*, vol. 4, no. 2, pp. 29–44, 2013.

[8] "Openai api," https://openai.com/blog/openai-api/, June 2020.

[9] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate Before Use: Improving Few-Shot Performance of Language Models," *arXiv e-prints*, p. arXiv:2102.09690, Feb. 2021.

[10] "unittest — unit testing framework — python 3.9.2 documentation," https://docs.python.org/3/library/unittest.html.

[11] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev, "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," *arXiv e-prints*, p. arXiv:1809.08887, Sep. 2018.

[12] R. M. Karampatsis, H. Babii, R. Robbes, C. Sutton, and A. Janes, "Big code != big vocabulary: Open-vocabulary models for source code," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, 2020, pp. 1073–1085.

[13] V. Raychev, M. Vechev, and E. Yahav, "Code completion with statistical language models," in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2014, pp. 419–428.

[14] X. Gu, H. Zhang, D. Zhang, and S. Kim, "Deep api learning," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016, pp. 631–642.

[15] E. Chatzikoumi, "How to evaluate machine translation: A review of automated and human metrics," *Natural Language Engineering*, vol. 26, no. 2, p. 137–161, 2020.

[16] T. Gao, A. Fisch, and D. Chen, "Making Pre-trained Language Models Better Few-shot Learners," *arXiv e-prints*, p. arXiv:2012.15723, Dec. 2020.