Recent Trend of Code Summarization*

1st Youngmi Park* 2nd Ahjeong Park† 3rd Chulyun Kim‡
Department of Information Technology Engineering
†Sookmyung Women's University
Seoul, South Korea
*yommi1121@sookmyung.ac.kr †ahjeong@sookmyung.ac.kr ‡cykim@sookmyung.ac.kr

***Abstract*—Code comments explain the operational process of a computer program and increase the long-term productivity of programing such as debugging and maintenance, and it is increasingly needed to develop methods that automatically generate natural language comments from programming codes. With the development of deep learning, various models that were excellent in the NLP domain are applied to the comment generation task. In recent studies, the performance is improved by simultaneously utilizing the lexical information of the code token and the syntactical information obtained from the syntax tree. In this paper, in order to improve the accuracy of automatic comment generatio**

I. Introduction

Code comments are the main factor for under standing the working process of source code. Code comments sometimes help developers save time in program maintenance and finding bugs. Research [1], [2] shows that commented code is easier to understand than uncommented code, and high-quality code comments can effectively improve program comprehension.

In general, during software development and maintenance, developers often spend a lot of time understanding the program. However, it takes a lot of time and effort to write own comments and keep them up to date [3]. And as the vast open source and software scale grows, the need for code comment generation technology that automatically generates high-quality natural language comments is increases.

With the development of deep learning, a variety of excellent models in the NLP domain have been applied to comment generation [4]. Examples of lexical information and syntax information used in recent studies to achieve SOTAs [4], [5]

II. Key Contribution

The ALSI-Transformer proposed in this paper uses two types of sequence information extracted from the source code. One is a code sequence that is a lexical expression, and the other is a new data type CAT that is a syntax expression.

Our contributions are listed as follows:

- **Contribution 1:** We propose a new data type, CAT(Code-Aligned Type Sequence), aligned according to the code sequence.
- **Contribution 2:** To the best of our knowledge, we are the first to align the order of syntactic representation to semantic representation.
- **Contribution 3:** As a result of comparing our model based on one encoder with two encoders, we observed that our model is 12.6% smaller in size and has excellent comment generation perfomance.
- **Contribution 4:** We evaluate the performance of several methods to create a combined feature that merges two inputs. As a result, it was confirmed that the performance of the Gate Network is the best, and this is applied to our model.
- **Contribution 5:** As a result of comparing with six baselines for the three performance metrics widely used in NMT, BLEU and METEOR, our model 132 achieved state-of-the-art with BLEU 53.80% and METEOR 66.11%.

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 (Project Name: Development of software copyright application technology for fair trade and distribution, Project Number: R2022020041, Contribution Rate: 90%)

III. Future Works and Conclusion

In this paper, we proposed Transformer based code comment generation model ALSI-Transformer. We improved the performance by further learning a syntactic representation. We also compressed the dimensions of the data to speed up the training process of neural networks with CNNs. In future work, One challenge for our model is to apply it to programming languages. Furthermore, to improve the performance of the proposed method, we plan to explore additional information available in addition to the syntactic information such as CAT that we utilize and apply other state-of-the-art deep learning models.

References

[1] G. Sridhara, E. Hill, D. Muppaneni, L. Pollock, and K. Vijay-Shanker, "Towards automatically generating summary comments for java methods," in *Proceedings of the IEEE/ACM international conference on Automated software engineering*, 2010, pp. 43–52.

[2] H. He, "Understanding source code comments at large-scale," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 1217–1219.

[3] F. Wen, C. Nagy, G. Bavota, and M. Lanza, "A large-scale empirical study on code-comment inconsistencies," in *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*. IEEE, 2019, pp. 53–64.

[4] X. Hu, G. Li, X. Xia, D. Lo, and Z. Jin, "Deep code comment generation," in *2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC)*. IEEE, 2018, pp. 200–20 010.

[5] Z. Li, Y. Wu, B. Peng, X. Chen, Z. Sun, Y. Liu, and D. Paul, "Setransformer: A transformer-based code semantic parser for code comment generation," *IEEE Transactions on Reliability*, 2022.