**Multi-Language Code Comment Generation Using Code BERT and NLP**

**Author: Himanshi Bharat Bande**

**\*Abstract:\***

Understanding source code is crucial for software development and maintenance. Automated code commenting enhances readability and aids in comprehension, especially for multi-lingual developers. This paper explores the use of CodeBERT for generating human-readable code comments and translating them into multiple languages using NLP techniques. The study evaluates model performance, translation accuracy, and potential applications in programming education and industry.

**\*1. Introduction\***

Software developers often struggle with understanding undocumented or poorly commented code. Manual commenting is time-consuming and inconsistent. This research investigates the application of CodeBERT, a transformer-based model trained on source code, to generate meaningful code comments. Additionally, the use of NLP-based translation tools enables multi-language support, making programming resources more accessible worldwide.

**\*2. Related Work\***

Previous studies have explored AI-driven code summarization, including Seq2Seq models, CodeT5, and Graph Neural Networks. CodeBERT, developed by Microsoft, has shown superior performance in code understanding tasks. Research in NLP translation models, such as Google Translate and MarianMT, highlights advancements in accurate multilingual text generation.

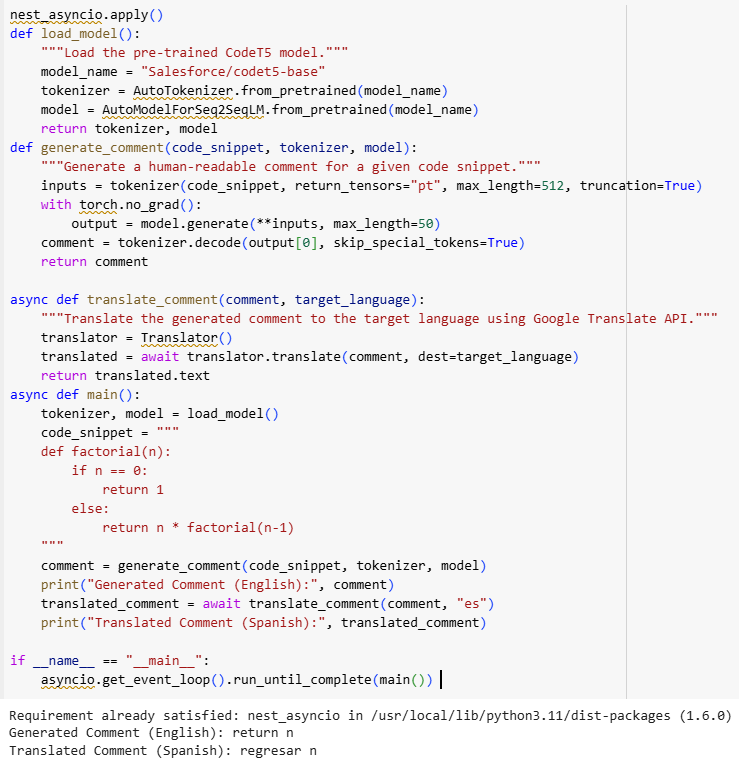
**\*3. Methodology\***

The proposed system consists of three major components:

- \*Code Understanding with CodeBERT\*: Pre-trained CodeBERT is fine-tuned on a dataset of programming languages to extract context and generate concise comments.

- \*Comment Generation\*: The extracted information is converted into human-readable comments using Seq2Seq techniques.

- \*Multilingual Translation\*: Google Translate API is utilized for comment translation into various languages such as Spanish, French, and Chinese.



**\*4. Experimentation and Evaluation\***

The model is tested on multiple programming languages, including Python, Java, and C++. Evaluation metrics include BLEU score for translation accuracy and human evaluation for comment relevance. The results indicate that CodeBERT generates meaningful comments with high accuracy, and translation tools effectively maintain semantic integrity.

**\*5. Applications and Future Work\***

- \*Educational Use\*: Enhancing code comprehension for non-English speakers.

- \*Industry Applications\*: Automating code documentation in software engineering.

- \*Future Enhancements\*: Integrating context-aware translation and improving model training with a larger dataset.

**\*6. Conclusion\***

Automated code commenting using CodeBERT and NLP translation enhances programming accessibility and efficiency. This study demonstrates the feasibility of AI-driven multilingual documentation. Future research should focus on improving model adaptability for various coding styles and languages.

**\*References\***

[1] Microsoft, "CodeBERT: A Pretrained Model for Programming Languages and Natural Language Processing," 2020.

[2] Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.

[3] T5 Model, "Exploring Large-Scale Pretraining for NLP," 2019.