Large Language Models for Code Comment Consistency Checking

Peter Elmers

Goal

Efficiently use large language models (LLMs) to detect comments that are inconsistent with code

Motivation

- Success of LLMs in many areas of natural language research
  - Including code generation, e.g. GitHub Copilot
- Insight: **83%** of professional developer time is spent on code navigation and understanding
  - Thus, the **development bottleneck** is reading, not writing!
- **Motive:** Improve code quality by detecting inconsistent comments

Research Questions

RQ1: Can we apply LLMs to classify comment consistency?

- **Approach**: model fine tuning

RQ2: How well can we generalise the model to languages other than Java?

- **87% of prior research** only studied Java!

RQ3: How can we create a benchmark that more closely matches real world situations?

- **Approach**: mine open source software repositories

RQ4: How do practitioners react to the results of this model in a social context?

- **Approach**: send pull request comments on GitHub

Feasibility Study Results

1 Epoch Model Test

Why?

- **Mitigate risks** that LLMs cannot learn the problem, or take too long to train

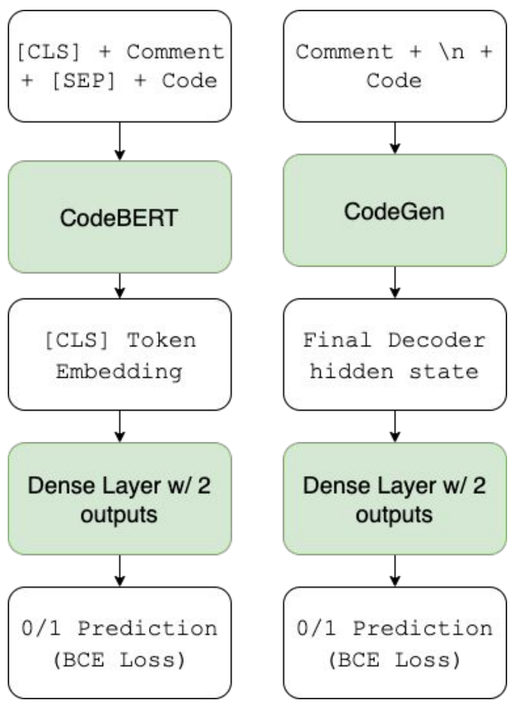
Setup

- Re-use **Java** dataset published in previous work
- Train CodeBERT and CodeGen-350M for **1 epoch**
- Default hyperpameter choices, no tuning done
- Training Time: **15-30 minutes** on free Colab

Results

Model Type	F1 Score
Liu et al. (Random forests)	0.655
Panthaplackel et al. (GNN-based)	0.706
Steiner et al. (BERT)	0.864*
Our CodeBERT	0.837
Our CodeGen	0.843

*: subset score not published, full set used



Case Study: Pull Request Comments

Why?

- Training dataset is mined, *may have false examples*
- Validate approach to **RQ3**, if a maintainer asks to "update comment" on a pull request, the comment is **probably inconsistent!**

Setup

- Script on GitHub API to download **>1 million** PR comments (*8 hours to execute!)*
- Filter text to "update/fix/outdate" + "comment"

Example

Result – method can be used, but **requires care**, does "comment" refer to source code or PR discussion?

Discussion

Knowledge Contributions

- **Potential**
  - New datasets to spur multi-lingual research (**RQ2)**
  - Curated benchmark set (**RQ3)** to validate real-world use
- **Implementation**
  - Reduce technical debt, greater productivity (**RQ4**)

Ethics

- Avoids many ethical risks of *generative models*
  - Reproduction of licensed code
  - Existential threat to job security
- **Risk Compensation**
  - If developers rely entirely on this consistency checker, will they compensate with carelessness?