AUTHORS

Paris Loizides
p.loizides@student.tudelft.nl

SUPERVISORS

Prof. Dr. Arie van Deursen
Assistant Prof. Dr. Maliheh Izadi
ir. Jonathan Katzy

LLM of Babel: Evaluation of LLMs on code for non-English use-cases

A Thesis Submitted to EEMCS Faculty Delft University of Technology

TU Delft Delft University of Technology

01. Introduction

Large Language Models (LLMs) like GPT and BERT have:

- Revolutionized **software development** by significantly enhancing coding efficiency.
- Shown broader **educational benefits.**

Despite these advancements:

- Performance disparity exists in non-English programming environments.

→ limits the ***global applicability*** of such technologies.

This research seeks to address this gap by examining how LLMs perform across Java **code summarization tasks** when applied to non-English languages, with a particular focus on the **Greek language** on **StarCoder 2**.

02. Research Questions:

Our study makes several key contributions:

RQ1: What types of errors are most common in Greek and other non-English languages, and how can a hierarchical error taxonomy help guide future developments in LLM technology?

RQ2: How does the tokenization process of prompts affect the performance of LLMs in recognizing Greek and generating comments?

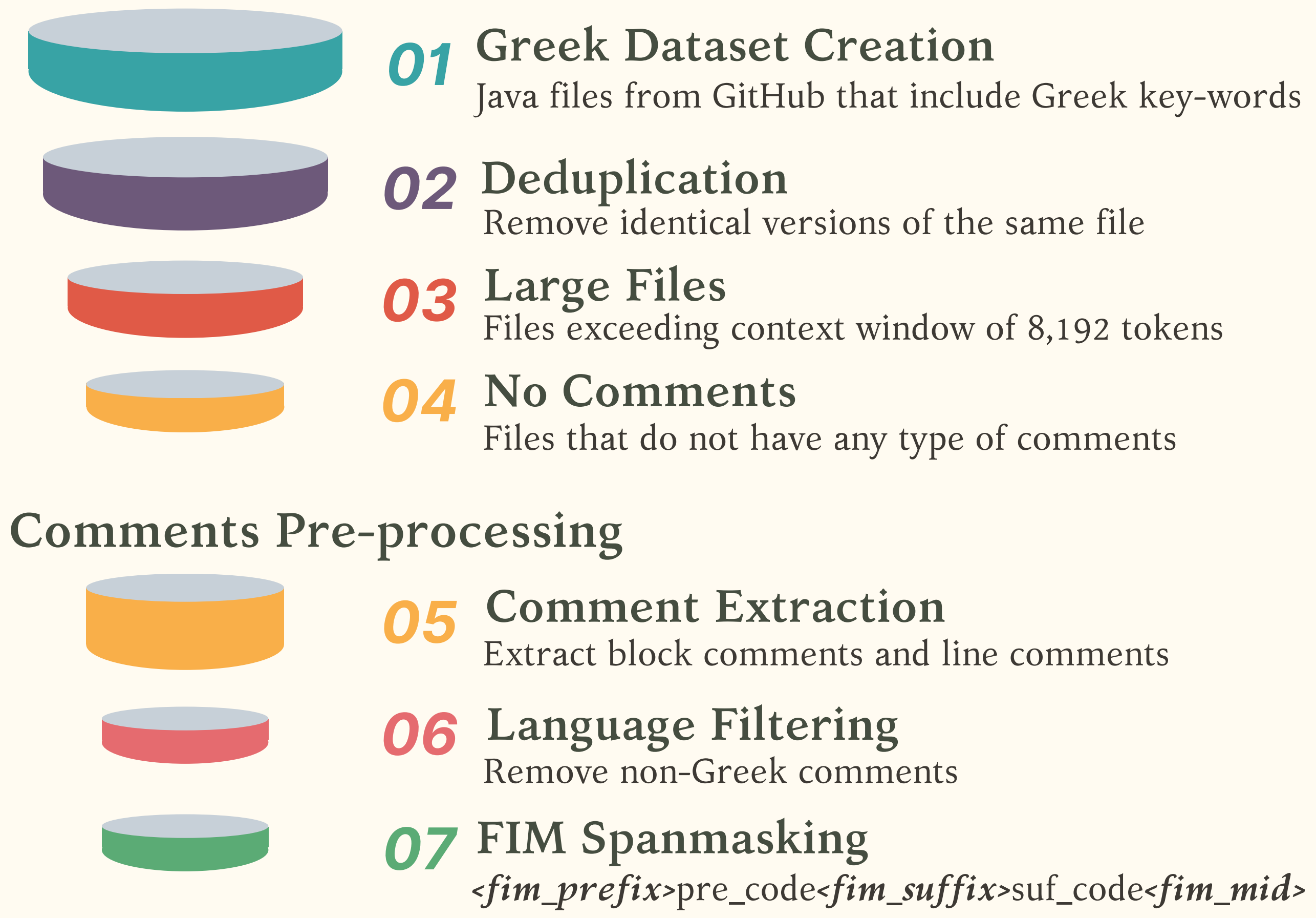
RQ3: What is the quantitative performance of StarCoder 2 in code summarization when prompted with Greek-documented code snippets?

03. Methodology

Model:

StarCoder 2 was selected for its high performance in code-related tasks and its training using the **Fill-in-the-Middle (FIM)** objective.

Dataset Filtering



01 Greek Dataset Creation
Java files from GitHub that include Greek key-words

02 Deduplication
Remove identical versions of the same file

03 Large Files
Files exceeding context window of 8,192 tokens

04 No Comments
Files that do not have any type of comments

Comments Pre-processing

05 Comment Extraction
Extract block comments and line comments

06 Language Filtering
Remove non-Greek comments

07 FIM Spanmasking
<***fim_prefix***>pre_code<***fim_suffix***>suf_code<***fim_mid***>

Open Coding

- Develop **Hierarchical Error Taxonomy** in collaboration with research team using **open coding approach**.
- Identify most **frequent errors** for the **Greek** language using StarCoder 2.
- Qualitatively analyse data.

Tokenization Experiment

Compare information density of **3 distinct tokenizers** on Greek comments to determine the effect of **Mathematical documents** on tokenization and training of LLMs.

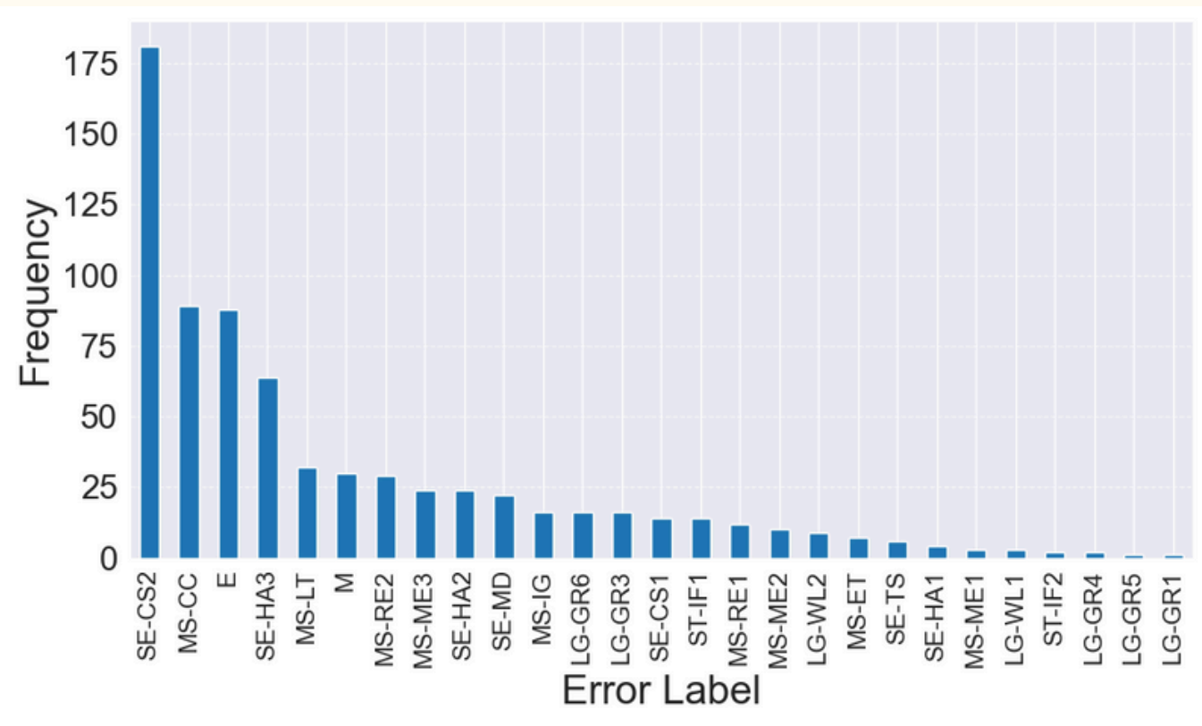
Tokenizers:

- StarCoder 2
- Meltemi-7B-v1: first **Greek** LLM
- OpenWebMath Dataset **Custom** Tokenizer

Quantitative Analysis

1) **Accuracy Rate** (Ability of model to correctly summarize code snippet)
2) **BLEU** Score (Bilingual Evaluation Understudy)
3) **ROUGE** Score (Recall-Oriented Understudy for Gisting Evaluation)
4) **Semantic Similarity** (using Multilingual Sentence Transformers)

04. Results - Taxonomy

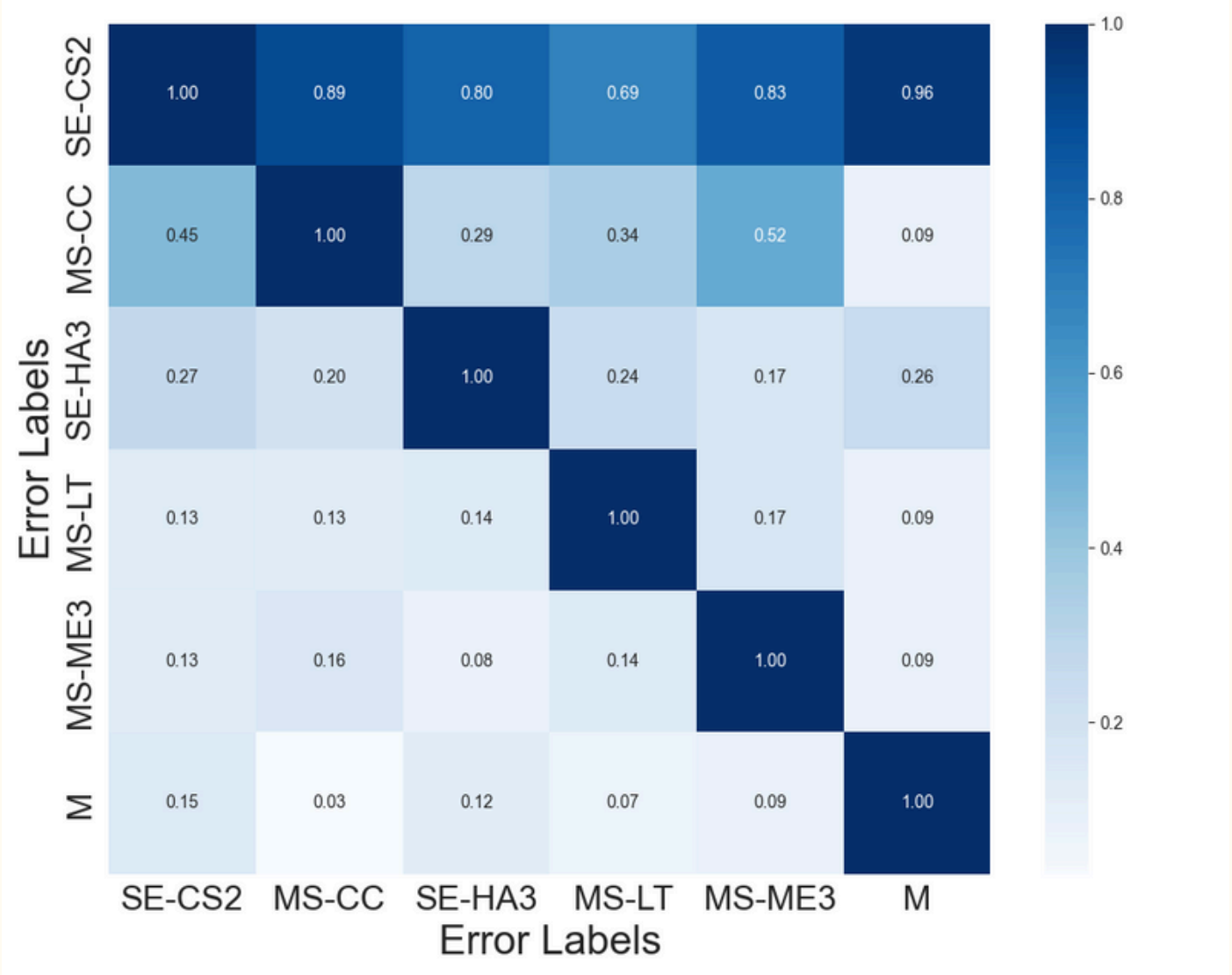


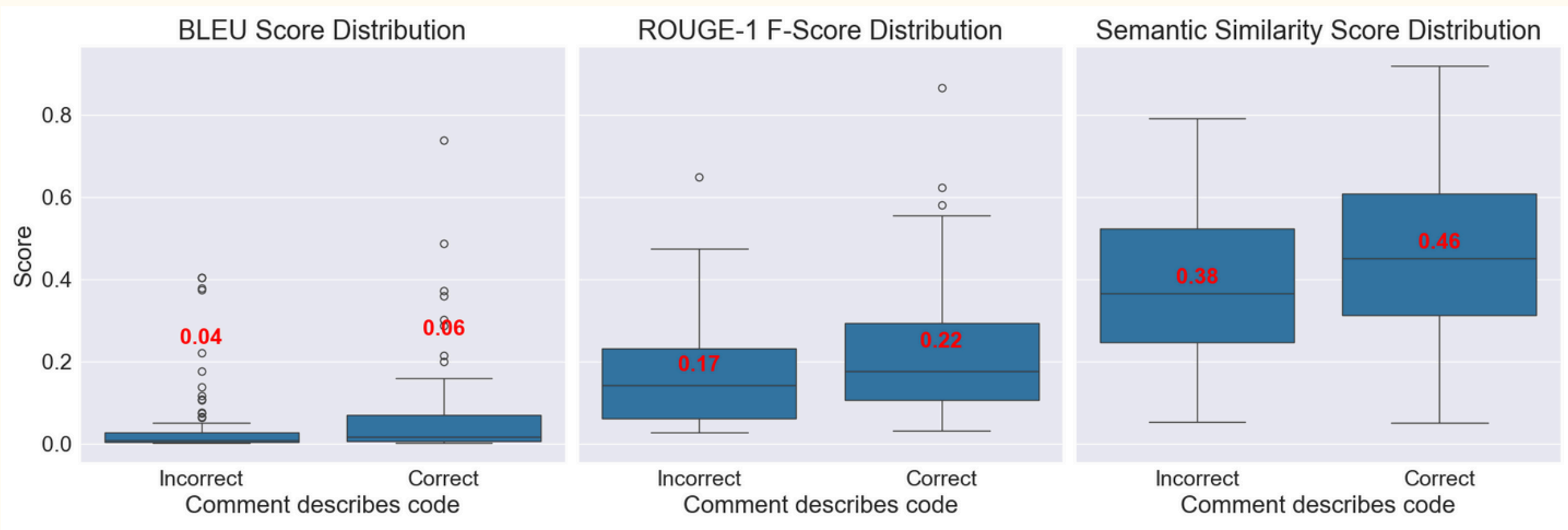
Most common Errors:

181	Code To run
89	Copying Context
88	Excluded
64	Educated Guess
32	Late termination
29	Verbatim Repetition
24	Memorization

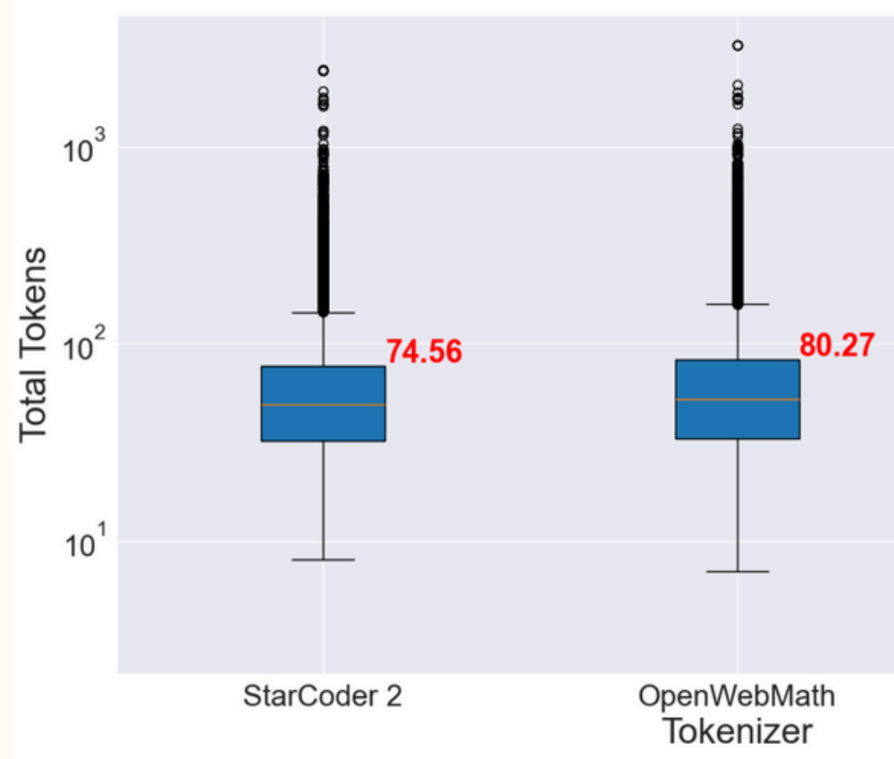
Significant Co-occurance:

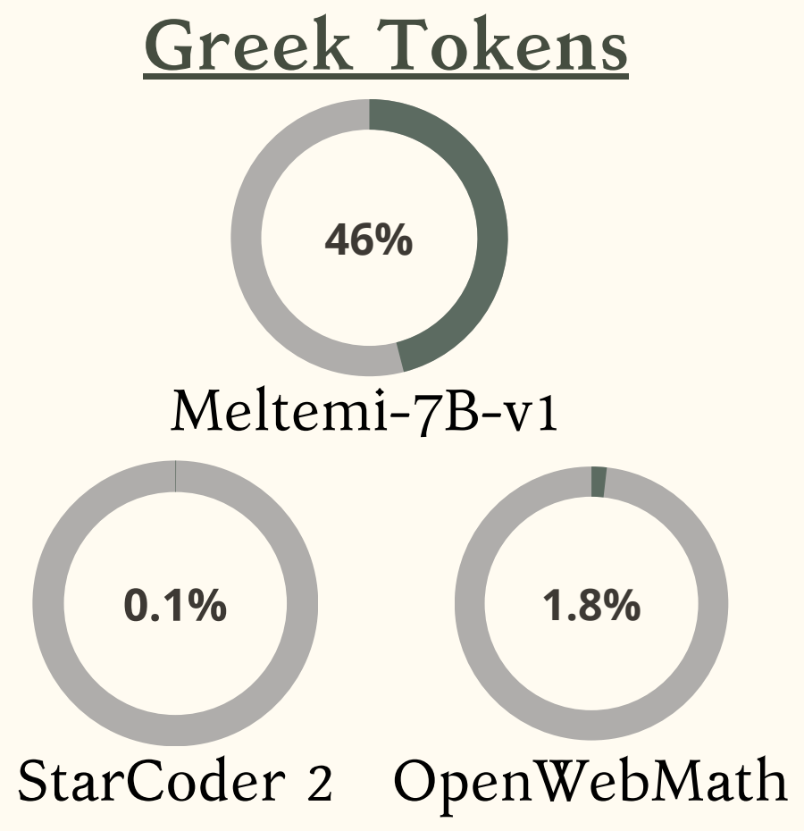
- **(x, SE-CS2)**: >69%
  - all labels highly co-occur with code generation
- **(SE-CS2, MS-CC)**: 45%
  - code snippet generation co-occurrance with copying context from file
- **(SE-CS2, SE-HA3)**: 27%
  - code snippet generation co-occurrance with hallucination grounded to context
- **(MS-ME3, MS-CC)**: 52%
  - when memorization occurs, generation includes copied context
  - showcases **overfitting** on Greek documented files due to limited available resources and forking of Greek repositories





05. Results - Tokenization





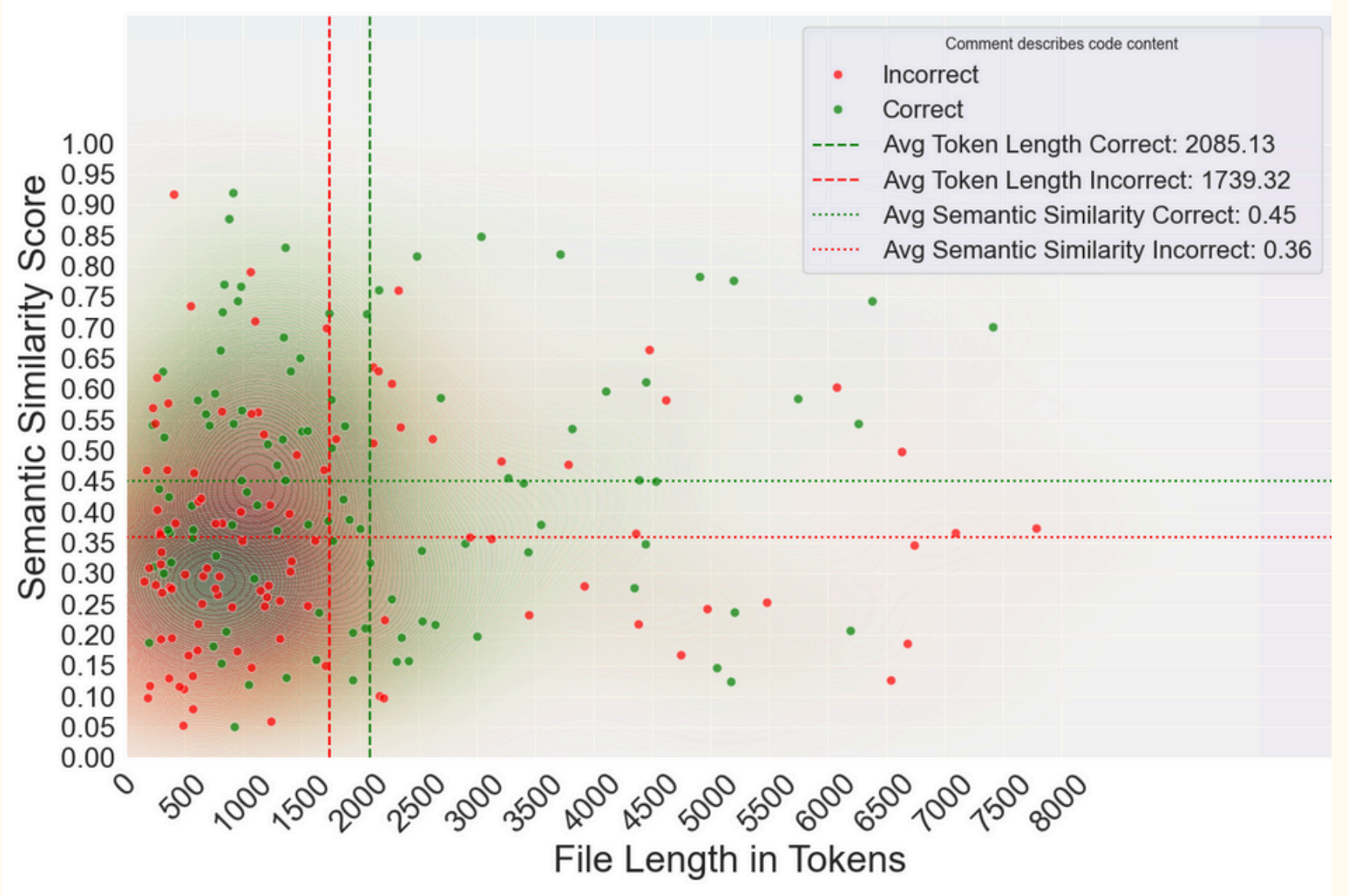
- **StarCoder 2 Vs OpenWebMath**
  - Similar total tokens average
  - → Low information density of individual letter tokenization
- **StarCoder 2 Vs Meltemi-7B-v1**
  - **3x** better information density of Greek tokenizer

06. Results - Quantitative

- **Kernel Density Estimation (KDE)** shows the effect of context length

Correct Predictions: *Semantic Similarity*: 0.5
File token length: 1000 tokens

Incorrect Predictions: *Semantic Similarity*: 0.3
File token length: 500 tokens



- **Accuracy Rate:**
  - Manual labelling yields **49%** of predictions to correctly describe code.
- **Effectiveness of Evaluation Metrics:**
  - **Semantic Similarity** best differentiates among Correct and Incorrect predictions showing a **larger gap, uniform distributions** and **no outliers.**

07. Conclusion and Future Work

- Identified the **most common errors** for the Greek language using the established **hierarchical error taxonomy.**
- **Tokenization** experiments provided insights into **training data** and **tokenizer design**.
- Found **Semantic Similarity** metric **more effective** than BLEU and ROUGE for multilingual evaluation.
- Future research should **refine taxonomy** by extending research to **Greeklish and other languages** and analyze **The Stack v2's Greek corpus** for a better understanding of the Greek language in coding environments.