Evaluating CodeGemma-7B for Dutch Code Comment Generation

Author

Sander Vermeulen
s.r.vermeulen@student.tudelft.nl

Supervisors

Maliheh Izadi
Responsible Professor
m.izadi@tudelft.nl

Arie van Deursen
Responsible Professor
arie.vandeursen@tudelft.nl

Jonathan Katzy
Supervisor
j.b.katzy@tudelft.nl

1 Introduction

Large Language Models (LLMs) are becoming an increasingly effective tool for code completion [1,2]. However, questions arise regarding the **effectiveness beyond English**, the language most models are trained on [1]. This paper investigates the adaptability of LLMs when **used outside of their intended use**, particularly focussing on the model CodeGemma-7B [3] when prompted with Dutch context.

2 Aim

The leading question of the paper is: **How effective is the CodeGemma-7B model in generating code-comments for programming in Dutch?**
To answer this question, we investigate the following research questions:

RQ1 What kind of errors does the model make?

RQ2 How well does the generated output match the original according to ROUGE [4] and BLEU [5] metrics?

RQ3 What kind of inefficiencies exist in the tokenizer used by the model?

3 Methodology

Data Preparation:

- **Collect source code dataset:** The dataset used consists of open-source Java code containing one of the top 2500 most common Dutch words found using the GitHub search API.
- **Mask out existing comments:** Utilized regular expressions to locate and mask existing comments. Retained the first two words for line comments and three words for block comments of the original comment to ensure sufficient context for Dutch language prediction.
- **Add LLM delimiters for fill-in-the-middle (FIM):** Added the prefix, middle, and suffix delimiters for the model.
- **Random selection:** Ensured only one comment was included per repository to avoid skewed data.

Inference:

- **Tokenize prompt:** Tokenize the masked data.
- **Run the model:** Ran inference on the tokenized data to generate new comments.
- **Run metrics:** We used the ROUGE-L and BLEU-4 metrics to compare the original and generated comments.

4 Results

Figure 1:: Distribution of evaluated comments

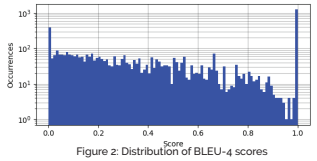

Figure 2: Distribution of BLEU-4 scores

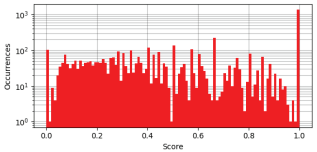

Figure 3: Distribution of ROUGE-L scores

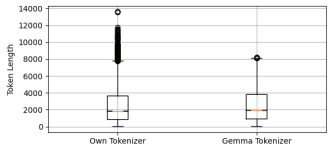

Figure 4: Custom-trained tokenizer vs Gemma tokenizer

Failure category plus label ID	Count
Model Errors	**611**
Model Specific	234
(MS-IG) Incoherent Generation	1
(MS-CC) Copy Context	33
Memorization	36
(MS-ME1) Contains PII	35
(MS-ME2) Contains URL	0
(MS-ME3) Verbatim Memorization	1
(MS-ET) Early Termination	11
(MS-LT) Late Termination	109
Repetition	44
(MS-RE1) Pattern Repetition	18
(MS-RE1) Verbatim Repetition	26
Linguistic	81
Grammar	64
(LG-GR1) Plurality	1
(LG-GR2) Conjugation	1
(LG-GR3) Gendering	3
(LG-GR4) Spelling	35
(LG-GR5) Capitalization	1
(LG-GR6) Cohesion	23
(LG-IS) Usage of Incorrect Synonym	1
Wrong Language	16
(LG-WL1) Undesired Translations	9
(LG-WL2) Incorrect Language	7
Semantic	285
(SE-MD) Missing Details	32
(SE-TS) Too Specific	3
Hallucination	159
(SE-HA1) Misplaced Facts	16
(SE-HA2) Contextual Discrepancy	5
(SE-HA3) Educated Guess	138
Code Snippet Inclusion	91
(SE-CS1) Commented Out Code	10
(SE-CS2) Code Intended to Run	81
Syntax	11
Incorrect Comment Format	11
(ST-IF1) Style Inconsistency	10
(ST-IF2) Omitted Identifier	1
Accepted Comments	**449**
Excluded Comments	**356**

Table 1: Taxonomy of failure categories

Qualitative results showed the model's promising performance in generating Dutch code comments, with 53% of non-excluded comments being accepted as displayed in Figure 1. The most common errors we found were related to educated guesses and late terminations. The full taxonomy of all labels we assigned is displayed in Table 1. Quantitative results showed that our results are very similar 26% of the time according to BLEU-4 and 28% according to ROUGE-L. The distribution of the scores is displayed in Figure 2 and 3. Finally, training our own tokenizer showed a mean of 5.06% lower as displayed in Figure 4, with the compression factor being 5.35% better.

5 Conclusion

This study examined the potential and challenges of using the CodeGemma-7B model for generating comments, showing its **promise in natural language and code generation**. However, issues such as **producing incorrect statements** and **late terminations** indicate the need for improvements. These issues can be improved through better training processes, bias mitigation, and standardized benchmarks Additionally, our **custom tokenizer outperformed** the Gemma tokenizer, showing the importance of efficient tokenization in non-English languages.

References

[1] M. Izadi, R. Gismondi, and G. Gousios, "Codefill: multi-token code completion by jointly learning from structure and naming sequences," in Proceedings of the 44th International Conference on Software Engineering, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 401–412. Available: https://doi.org/10.1145/3510003.3510172

[2] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," ACM Trans. Knowl. Discov. Data, vol. 18, no. 6, apr 2024. Available: https://doi.org/10.1145/3649506

[3] CodeGemma Team, A. J. Hartman, A. Hu, C. A. Choquette-Choo, H. Zhao, J. Fine, J. Hui, J. Shen, J. Kelley, J. Howland, K. Bansal, L. Vilnis, M. Wirth, N. Nguyen, P. Michel, P. Choy, P. Joshi, R. Kumar, S. Hashmi, S. Agrawal, S. Zuo, T. Warkentin, and Z. e. a. Gong, "Codegemma: Open code models based on gemma," 2024. Available: https://goo.gle/codegemma

[4] 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, P. Isabelle, E. Charniak, and D. Lin, Eds. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. Available: https://aclanthology.org/P02-1040

[5] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. Available: https://aclanthology.org/W04-1013

TU Delft