Cross-lingual Code Clone Detection: When LLMs Fail Short Against Embedding-based Classifier

Micheline B. MOUMOULA
micheline.moumoula@uni.lu
University of Luxembourg
Luxembourg

Abdoul K. KABORE
abdoulkader.kabore@uni.lu
University of Luxembourg
Luxembourg

Jacques KLEIN
jacques.klein@uni.lu
University of Luxembourg
Luxembourg

Tegawendé F. BISSYANDE
tegawende.bissyande@uni.lu
University of Luxembourg
Luxembourg

ABSTRACT

Cross-lingual code clone detection has gained attention in software development due to the use of multiple programming languages. Recent advances in machine learning, particularly Large Language Models (LLMs), have motivated a reexamination of this problem.

This paper evaluates the performance of four LLMs and eight prompts for detecting cross-lingual code clones, as well as a pre-trained embedding model for classifying clone pairs. Both approaches are tested on the XLCoST and CodeNet datasets.

Our findings show that while LLMs achieve high F1 scores (up to 0.98) on straightforward programming examples, they struggle with complex cases and cross-lingual understanding. In contrast, embedding models, which map code fragments from different languages into a common representation space, allow for the training of a basic classifier that outperforms LLMs by approximately 2 and 24 percentage points on the XLCoST and CodeNet datasets, respectively. This suggests that embedding models provide more robust representations, enabling state-of-the-art performance in cross-lingual code clone detection.

KEYWORDS

Cross-Language Pairs, Code Clone Detection, Large Language Model, Prompt Engineering, Embedding Model

ACM Reference Format:
Micheline B. MOUMOULA, Abdoul K. KABORE, Jacques KLEIN, and Tegawendé F. BISSYANDE. 2024. Cross-lingual Code Clone Detection: When LLMs Fail Short Against Embedding-based Classifier. In *39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24), October 27-November 1, 2024, Sacramento, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3691620.3695335

1 INTRODUCTION

Code clone detection is a significant challenge in software development, with studies estimating that 5% to 23% of clones exist in a software system [8], and Type-4 clones being the most difficult to detect [3]. While clone detection is commonly done within a single language, modern software often integrates multiple languages [7], requiring cross-lingual clone detection. Collaborative development across languages increases the complexity, as changes in one language must be mirrored in others, making the process resource- and time-intensive. An automatic system for detecting clones across languages is essential for managing cross-lingual systems efficiently [9].

The literature includes several approaches and tools for cross-language code clone detection [2, 4–6, 9, 11, 12, 15]. Most of them rely on machine learning techniques to capture the syntactic and semantic relationships between different parts of the source code. With the recent rise of Large Language Models (LLMs) and their ability to understand and generate human-quality text, LLMs offer a promising avenue for tasks such as code comprehension and analysis.

This work explores the effectiveness of LLMs and Embedding Models (EMs) for cross-lingual code clone detection. Using the two widely adopted datasets, we evaluated the performance of four LLMs (Falcon-7B-Instruct [1], LLAMA2-Chat-7B [13], Starchat-β [14], and GPT-3.5-Turbo [1]) under various prompting strategies across eleven programming languages. Our second exploration leverages Text-Embedding-Ada-002, an embedding model from OpenAI to generate vector representations of code fragments. We then compute the cosine similarity between the vectors to determine their similarity. Additionally, we trained custom binary classifiers on the generated embeddings to further enhance clone detection accuracy.

2 METHODOLOGY

This section outlines the experimental methodology employed to evaluate the performance of LLMs and classification models.

❶ **Cross-lingual code clone detection as an NLP task**. This research explores the potential of LLMs for cross-lingual code clone detection using prompt engineering. We developed eight prompts designed to elicit either a binary "yes/no" response or a similarity score, aiming to assess LLMs' performance in this task. Through experimentation, we evaluate their ability to identify cross-lingual

ASE '24, October 27-November 1, 2024, Sacramento, CA, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1248-7/24/10
https://doi.org/10.1145/3691620.3695335

[1] https://openai.com/

code clones based on semantic analysis. Our findings offer valuable insights into the effectiveness of LLMs in addressing this key software engineering challenge.

❷ **Cross-lingual code clone detection as a Classification task**. To evaluate traditional machine learning models, we replicated Keller et al.'s approach using the "Text-embedding-Ada-002" model. We applied two basic classifiers k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) to categorize code fragments based on these embeddings. Additionally, we explored a direct similarity-based method by computing cosine similarity between cross-lingual code fragments in a unified embedding space. We systematically adjusted the similarity threshold to optimize clone pair identification performance.

2.1 EXPERIMENTAL SETUP

Using two widely accepted datasets, XLCoST [16] and CodeNet [10], this study explores four key research questions: the impact of prompt engineering on improving LLMs for cross-lingual code clone detection, the extent of LLMs' understanding of this task, the influence of programming language similarity on LLM performance, and whether LLMs outperform traditional classification models in cross-lingual code clone detection. All performances are evaluated using the metrics precision, recall, and F1-score.

3 RESULTS

❶ **LLMs Performance on Cross-Lingual Code Clone Detection and Impact of Prompt Engineering.**
In general, LLMs can detect cross-language code clones and GPT-3.5-Turbo outperforms all of them. Combining all of the results for this section GPT-3.5-Turbo got an F1 score of 0.98 and 0.59 on the XLCoST and the CodeNet datasets respectively.

❷ **LLMs' Reasoning for Cross-Lingual Code Clone Detection.**
Our qualitative analysis revealed that Falcon-Instruct-7B and LLAMA2-Chat-7B tend to overclassify code pairs as clones, leading to high false positive rates. In contrast, Starchat-β frequently misclassifies clones as non-clones, resulting in a high false negative rate, likely due to its difficulty reasoning in cross-lingual contexts. Overconfidence in some models, such as GPT-3.5-turbo, was noted, with outputs claiming that "code snippets in different languages cannot be clones." To address these issues, we designed a prompt focusing on "overall structure and logic." This led to a significant F1 score improvement, with Starchat-β and LLAMA2-Chat-7B showing gains of 27 to 48 percentage points.

❸ **Influence of the Programming Languages Syntactical Similarity on LLMs Performances.** We observed a 10 percentage point F1 score gap between Java-C# and Java-Python fragments, reflecting Java-C#'s syntactic similarity. However, complex prompts with reasoning instructions helped reduce this gap, even for distinct language pairs like Java-PHP.

❹ **Traditional classification *vs.* LLMs.** Our results show that the Text-embedding-Ada-002 model can generate robust cross-lingual code representations, enabling effective clone detection using basic similarity measures or learned classification. Surprisingly, these traditional methods outperform LLMs with complex prompts by ~2 and ~24 percentage points on the XLCoST and CodeNet datasets, respectively. This suggests that the key challenge in cross-lingual code clone detection is creating a unified representation space for different programming languages, rather than focusing on advanced reasoning capabilities.

REFERENCES

[1] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance. (2023).

[2] Xiao Cheng, Zhiming Peng, Lingxiao Jiang, Hao Zhong, Haibo Yu, and Jianjun Zhao. 2017. *CLCMiner*: Detecting Cross-Language Clones without Intermediates. *IEICE Trans. Inf. & Syst.* E100.D, 2 (2017), 273–284. https://doi.org/10.1587/transinf.2016EDP7334

[3] Shihan Dou, Junjie Shan, Haoxiang Jia, Wenhao Deng, Zhiheng Xi, Wei He, Yueming Wu, Tao Gui, Yang Liu, and Xuanjing Huang. 2023. Towards Understanding the Capability of Large Language Models on Code Clone Detection: A Survey. http://arxiv.org/abs/2308.01191

[4] Yangkai Du, Tengfei Ma, Lingfei Wu, Xuhong Zhang, and Shouling Ji. 2024. AdaCCD: Adaptive Semantic Contrasts Discovery Based Cross Lingual Adaptation for Code Clone Detection. http://arxiv.org/abs/2311.07277

[5] Yong Fang, Fangzheng Zhou, Yijia Xu, and Zhonglin Liu. 2023. TCCCD: Triplet-Based Cross-Language Code Clone Detection. *Applied Sciences* 13, 21 (Nov. 2023), 12084. https://doi.org/10.3390/app132112084

[6] Mohamad Khajezade, Jie JW Wu, Fatemeh Hendijani Fard, Gema Rodríguez-Pérez, and Mohamed Sami Shehata. 2024. Investigating the Efficacy of Large Language Models for Code Clone Detection. http://arxiv.org/abs/2401.13802

[7] Pavneet Singh Kochhar, Dinusha Wijedasa, and David Lo. 2016. A Large Scale Study of Multiple Programming Languages and Code Quality. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, Suita, 563–573. https://doi.org/10.1109/SANER.2016.112

[8] Rainer Koschke. [n. d.]. Survey of Research on Software Clones. ([n. d.]).

[9] Kawser Wazed Nafi, Tonny Shekha Kar, Banani Roy, Chanchal K. Roy, and Kevin A. Schneider. 2019. CLCDSA: Cross Language Code Clone Detection using Syntactical Features and API Documentation. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, San Diego, CA, USA, 1026–1037. https://doi.org/10.1109/ASE.2019.00099

[10] Ruchir Puri, David S. Kung, Geert Janssen, Wei Zhang, Giacomo Domeniconi, Vladimir Zolotov, Julian Dolby, Jie Chen, Mihir Choudhury, Lindsey Decker, Veronika Thost, Luca Buratti, Saurabh Pujar, Shyam Ramji, Ulrich Finkler, Susan Malaika, and Frederick Reiss. 2021. CodeNet: A Large-Scale AI for Code Dataset for Learning a Diversity of Coding Tasks. http://arxiv.org/abs/2105.12655

[11] Nikita Sorokin, Dmitry Abulkhanov, Sergey Nikolenko, and Valentin Malykh. 2023. CCT-Code: Cross-Consistency Training for Multilingual Clone Detection and Code Search. http://arxiv.org/abs/2305.11626

[12] Chenning Tao, Qi Zhan, Xing Hu, and Xin Xia. 2022. C4: contrastive cross-language code clone detection. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. ACM, Virtual Event, 413–424. https://doi.org/10.1145/3524610.3527911

[13] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. http://arxiv.org/abs/2307.09288

[14] Lewis Tunstall, Nathan Lambert, Nazneen Rajani, Edward Beeching, Teven Le Scao, Leandro von Werra, Sheon Han, Philipp Schmid, and Alexander Rush. 2023. Creating a Coding Assistant with StarCoder. *Hugging Face Blog* (2023).

[15] Tijana Vislavski, Gordana Rakic, Nicolas Cardozo, and Zoran Budimac. 2018. LICCA: A tool for cross-language clone detection. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, Campobasso, 512–516. https://doi.org/10.1109/SANER.2018.8330250

[16] Ming Zhu, Aneesh Jain, Karthik Suresh, Roshan Ravindran, Sindhu Tipirneni, and Chandan K. Reddy. 2022. XLCoST: A Benchmark Dataset for Cross-lingual Code Intelligence. http://arxiv.org/abs/2206.08474