Exploring Existing Machine Learning Approaches Discerning the Quality of Source Code Identifiers

Marcel Simader
AI for Software Engineering, Topic 3
k11823075 / SKZ 521
marcel.simader@jku.at

Melissa Frischherz
AI for Software Engineering, Topic 3
k12011649 / SKZ 521
melissa.frischherz@gmail.com

***Abstract*—The quality of source code identifiers has a direct, and measurable impact on program comprehension, indirectly influencing validity, maintainability, and security of software. This paper proposes a small-scale survey of existing approaches assessing the quality of identifiers. Such a measure facilitates the evaluation of naming conventions in big code bases, suggestions of context-aware variable, method, or type names, or even finding semantic relationships between identifiers across languages and programs.**

By leveraging promising advancements in natural language processing (NLP) and machine learning, particularly the recent large language models (LLM), to analyze the information found in identifiers, we could greatly improve the code analyst's toolkit.

***Index Terms*—Machine Learning, Natural Language Processing, Code design, Maintainability, Software Quality/SQA.**

I. Introduction

***TODO*: Explain kinds of identifiers, and why they matter. Rich information content of atomic parts of a programming language's AST.**

***TODO*: State (and show evidence for) importance of good naming for quality assurance of software, maintainability, etc.**

II. Background

A. Machine Learning

***TODO*: Brief discussions of supervised/unsupervised techniques, neural networks, deep learning, and RNNs.**

B. Natural Language Processing

***TODO*: Brief discussions of n-grams, grammar, and the attention mechanism.**

III. Survey

A. Traditional Heuristic Approaches

***TODO*: "Intelligent" refactoring and renaming, e.g. in IntelliJ IDEA.**

B. RNNs and LSTMs

***TODO*: Literature like the LSTM approach of "A Neural Model for Method Name Generation from Functional Description" by Sa Gao et al. [1]**

C. LLMs and Beyond

***TODO*: Literature like the LLM approach of "How Well Can Masked Langauge Models Spot Identifiers That Violate Naming Guidelines?" by Johannes Villmow et al. [2]**

IV. Discussion and Future Work

***TODO*: Potential for fully-integrated tools, or CI pipeline utilities. Application of cutting-edge large language models. Surprising lack of experiments with GPTs?**

V. Conclusion

TODO

References

[1] S. Gao, C. Chen, Z. Xing, Y. Ma, W. Song, and S.-W. Lin, "A neural model for method name generation from functional description," in *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, Feb 2019, pp. 414–421. [Online]. Available: https://ieeexplore.ieee.org/document/8667994

[2] J. Villmow, V. Campos, J. Petry, A. Abbad-Andaloussi, A. Ulges, and B. Weber, "How well can masked language models spot identifiers that violate naming guidelines?" in *2023 IEEE 23rd International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2023, pp. 131–142. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10356698

[3] Y. Ju, Y. Tang, J. Lan, X. Mi, and J. Zhang, "A cross-language name binding recognition and discrimination approach for identifiers," in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2023, pp. 948–955. [Online]. Available: https://ieeexplore.ieee.org/document/10123604

[4] S. Butler, M. Wermelinger, Y. Yu, and H. Sharp, "Exploring the influence of identifier names on code quality: An empirical study," in *2010 14th European Conference on Software Maintenance and Reengineering*, 2010, pp. 156–165. [Online]. Available: https://ieeexplore.ieee.org/document/5714430

[5] M. Allamanis, E. T. Barr, C. Bird, and C. Sutton, "Suggesting accurate method and class names," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015. New York, NY, USA: Association for Computing Machinery, 2015, p. 38–49. [Online]. Available: https://doi.org/10.1145/2786805.2786849

[6] J. Shi, Z. Yang, J. He, B. Xu, and D. Lo, "Can identifier splitting improve open-vocabulary language model of code?" in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 1134–1138.

[7] A. Peruma, M. W. Mkaouer, M. J. Decker, and C. D. Newman, "Contextualizing rename decisions using refactorings and commit messages," in *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2019, pp. 74–85.

[8] C. D. Newman, A. Preuma, and R. AlSuhaibani, "Modeling the relationship between identifier name and behavior," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2019, pp. 376–378.

[9] C. Charitsis, C. Piech, and J. Mitchell, “Assessing function names and quantifying the relationship between identifiers and their functionality to improve them,” in *Proceedings of the Eighth ACM Conference on Learning @ Scale*, ser. L@S ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 291–294. [Online]. Available: https://doi.org/10.1145/3430895.3460161

[10] J. Zhang, J. Luo, J. Liang, L. Gong, and Z. Huang, “An accurate identifier renaming prediction and suggestion approach,” *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 6, sep 2023. [Online]. Available: https://doi.org/10.1145/3603109

Appendix

Additional Notes

```
Formulation of Topic Keywords/Key Phrases
------------------------------------------------------------------------------------

Topic is ‘AI in Software Engineering‘:
    - Natural Language Processing (NLP)
    - Large Language Models
    - Recurrent Neural Networks (RNN) and Encoder-Decoder models
    - Code quality measurement, improvement and assurance
    - Handling of identifiers in source code -- atomic units of knowledge, making up
      most of all source code
    - Understanding identifier in a functional sense (what do they mean to say?) and
      making (cross-language) connections between them (are they about the same
      thing?)

Research Question
------------------------------------------------------------------------------------

We answer the following questions[^1] in a maximum of two sentences, optimally one.
Statements here are not cited, as they are not part of the academic text. Beware of
plagiarism -- do not use anything from here directly.

Introduction:
> Identifiers make up about 70% of all source code, and hence encode a lot of
> information in a natural language format (with additional syntactic rules.) Code
> quality and comprehension might be dramatically improved by considering this
> information.

Problem Statement:
> Since identifiers are natural language, this information can be hard to make sense
> of with automated tooling for code analysis, improvement, automatic completion, or
> quality assurance.

State of the Art (in Literature):
> There exist NLP models, mostly based on deep learning but also some statistical
> analysis, which can predict identifiers to aid developers, measure the quality of
> existing identifiers, and establish semantic relationships between identifiers,
> within or across languages.

Our approach:
> We collect a list of literature presenting state-of-the-art machine learning
> techniques to draw broader conclusions from: Effectiveness of current approaches,
> categories of implementation, and possible pathways for future research.

Implementation:
> [ Not applicable? Err... we write the paper? ]

Results:
> Ideally, we present the reader with information which helps them understand the
> landscape of identifier-specific software engineering tools, and inspire them
> to find gaps in knowledge, or improve on existing ideas.

[^1]: Writing template, courtesy of Steve Eastbrook.
```