DOI:10.1145/3653323

Leveraging Language Models and Automatic Summarization in Online Programming Learning Environments

BY CARLOS ARECES, LUCIANA BENOTTI, FRANCO BULGARELLI, EMILIA ECHEVESTE, AND NADIA FINZI

THIS ARTICLE DESCRIBES the results of a collaborative project by researchers from the Universidad Nacional de Córdoba in Argentina and IKUMI SRL, an ed-tech company that provides massive-scale solutions for teaching programming. IKUMI develops educational resources for virtual spaces, including an open source platform called Mumuki,[a] which offers didactic paths with automated assessment of programming exercises.

For this project, IKUMI provided datasets with exercise solutions and discussions gleaned from a forum implemented on the Mumuki platform from 2021–2022. The aim is to collect feedback on programming language training programs for new software development professionals. The population was comprised of adults 18 years or older, with at least a high school education. It was evenly divided between women and men. Data from over 40,000 discussion threads, involving a total of more than 100,000 messages written in Spanish from various regions throughout Argentina, is currently being analyzed. The discussions and messages revolve around programming languages and introductory concepts, focused on imperative and object-oriented programming and exercises in programming languages such as JavaScript, Python, and Ruby.

The project set forth two overarching objectives (discussed briefly here):

Objective A. Enhance the interaction between tutors, the Mumuki platform, and the group of trainee programmers. By utilizing the stochastic language models of learners' errors in each programming language, training errors in the exercise are identified. This allows the student who needs help to read previous messages that helped a student with a similar error. The goal is to enable students to find answers more easily than the challenges they encounter in various programming exercises within the Mumuki platform. This involves providing better access to similar queries, thus reducing the time tutors are required to address the same obstacles. We defined a measure of distance between programs with errors so that those containing similar errors are considered close. To address this challenge, we used a language model commonly employed to determine when two natural language texts have similar meanings. In Moresi et al.,[10] we show that beginner programmers tend to mix the programming language they are learning with their native natural language, in this case, Argentinian Spanish. Therefore, using natural language models to represent programs is effective for programs written by beginners. A large proportion of the programs submitted by students do not compile, so techniques based on abstract syntax trees used previously are not useful.[7]

Teaching programming to students on the Mumuki platform.

PHOTO COURTESY OF MUMUKI.IO RESOURCES

The practices are part of the thinking and learning process, focusing on how to learn, and not just on what concepts are learned.

Objective B. Messages in the forum are summarized

[a] https://mumuki.io/home/

using automatic natural language techniques. These summaries are intended to help students identify the errors they are having and improve their ability to ask for help. When they use the selected recommendations, students will be provided with strategies to learn how to ask better questions iteratively. "Improving question formulation" represents acquiring a deeper understanding of the topic and emerges as a key strategy for advancing programming learning. The automatic summarization was implemented by adapting a technique known as TextRank[8] to the domain of the Mumuki forum in Spanish. We decided to use this technique due to two challenges of this project: First, Mumuki cannot rent a language model (such as ChatGPT) through an API because it compromises the privacy agreement they have with their users.[9] Second, the hardware and funding available for this project are quite limited so we had to choose a method that is not as computationally costly as full-fledged language models. Castillejo-Camacho[3] used a similar approach to fora in Spanish to predict course abandonment, but they did not do cheap automatic summarization, they did cheap sentiment analysis.

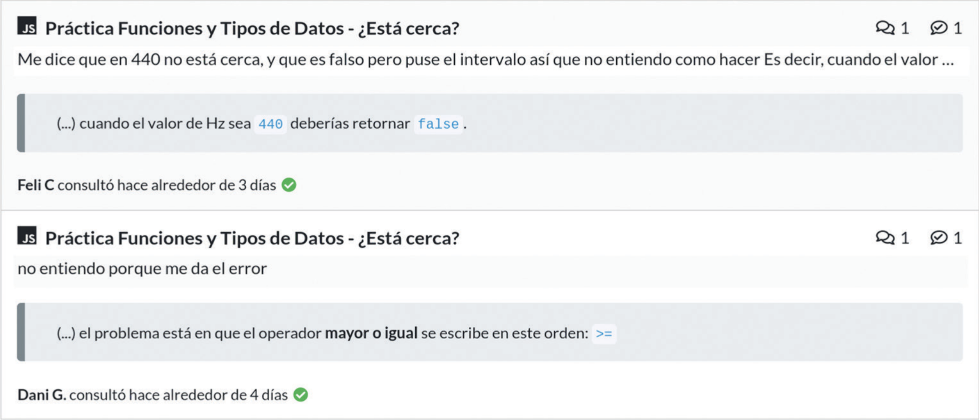
The final prototype has been evaluated with 5,000 online students with promising results. Statistics obtained from the use of the forum seems to indicate higher engagement and improved access to the available information. The successful implementation of this system holds the potential to enhance tutor-student interaction, optimize the use of resources, and foster effective learning experiences for students who speak Spanish and need help in their language. Also, this project is relevant to the current situation in Argentina in which the number of students interested in learning programming or in improving their programming skills by adding other languages has increased more than the number of teachers trained to offer help.

The accompanying figure illustrates the new interface of the Mumuki forum. Messages from students are ordered by how similar the code of the student asking for help is concerning the codes of the students that wrote the message. Below the student message—for example, "Me dice que 440 no está cerca . . ."—the automatic summarization of the tutor response is highlighted in gray—for example, "cuando el valor de Hz sea 440 deberías . . . ." It enables novice programmers to view summaries of conversations, empowering them to decide whether it is relevant to delve into a particular discussion within the Mumuki forum's search interface.

As a result of this work, the first version of a short textbook for new students was written, addressing the necessary computational practices to accompany students learning to program.[5] These practices develop as a person learns and engages with programming as well as programmer communities.[1] Some of these practices include planning, being incremental and iterative, copying and pasting responsibly, understanding errors, and asking questions in forums. They are part of the thinking and learning process, focusing on *how to learn*, and not just on what concepts are learned.

This collaborative project between Universidad Nacional de Córdoba and IKUMI leverages language models and automatic summarization to improve tutor-student interaction in the Mumuki platform. The innovative forum interface developed for Mumuki empowers novice programmers to make informed decisions, potentially transforming tutor-student dynamics.

Figure. This screen capture from the Mumuki forum shows two questions by students, with the most relevant question (according to the ranking provided by code proximity) automatically summarized here.



Práctica Funciones y Tipos de Datos - ¿Está cerca?

Me dice que en 440 no está cerca, y que es falso pero puse el intervalo así que no entiendo como hacer Es decir, cuando el valor ...

(...) cuando el valor de Hz sea `440` deberías retornar `false`.

Feli C consultó hace alrededor de 3 días ✓

Práctica Funciones y Tipos de Datos - ¿Está cerca?

no entiendo porque me da el error

(...) el problema está en que el operador **mayor o igual** se escribe en este orden: `>=`

Dani G. consultó hace alrededor de 4 días ✓

References

1. Brennan, K. and Resnick, M. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Assoc. (Vancouver, Canada, Apr. 2012)*, 25.
2. Burke, Q., O'Byrne, W.I., and Kafai, Y.B. Computational participation: Understanding coding as an extension of literacy instruction. *J. Adolescent & Adult Literacy 59*, 4 (2016), 371–375.
3. Castillejo-Camacho, D. Análisis del Abandono desde la perspectiva de los foros. *Tesis de Maestría Universitaria en Ingeniería y Ciencia de Datos*. Universidad Nacional de Educación a Distancia, España, 2022.
4. Chao, P.Y. Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education 95*, (2016), 202–215.
5. Echeveste, M.E., Bulgarelli, F., and Finzi, N. *Prácticas Computacionales Para Aprender a Programar: Una Guía Para Resolver Problemas de Programación*, 1a ed. Ciudad Autónoma de Buenos Aires, Mumuki, 2023.
6. Jacob, S.R. and Warschauer, M. Computational thinking and literacy. *J. Computer Science Integration. 1*, 1, (2018).
7. Karnalim, O. and Simon, S. Syntax trees and information retrieval to improve code similarity detection. In *Proceedings of the 22nd Australasian Computing Education Conf.* ACM, New York, NY, USA, 2020, 48–55; 10.1145/3373165.3373171
8. Kazemi, A., Pérez-Rosas, V., and Mihalcea, R. Biased TextRank: Unsupervised graph-based content extraction. In *Proceedings of the 28th Intern. Conf. Computational Linguistics*. ICCL, 2020, 1642–1652.
9. Mireshghallah, F. et al. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the Conf. Empirical Methods in Natural Language Processing*. ACL, 8332–8347.
10. Moresi, M., Gomez, M.J., and Benotti, L. Predicting students' difficulties from a piece of code. *IEEE Trans. Learning Technologies 14*, 3 (2021), 386–399.

Carlos Areces, Universidad Nacional de Córdoba, CONICET, Argentina.

Luciana Benotti, Universidad Nacional de Córdoba, CONICET, Argentina.

Franco Bulgarelli, Mumuki, Argentina.

Emilia Echeveste, Universidad Nacional de Córdoba, CONICET, Argentina.

Nadia Finzi, Mumuki, Argentina.