The paper titled "Extracting Features for Computational Thinking from Block-Based Code" by Param Damle and Glen Bull from the University of Virginia[1] proposes an automated evaluator (autograder) for Snap!, a block-based programming language. The autograder is designed to support undertrained instructors in introductory computer science classes. The novelty of this approach lies in its ability to work natively on a non-textual language and assess the computational thinking (CT) reflected in the structure of a student's submission rather than the accuracy or run-time of its execution[1].

The autograder assesses demonstrated knowledge of abstraction and iteration from an XML tree representation of a student's Snap! program[1]. The authors explore how to tailor feature extraction to capture CT elements, including consecutive repetition and encapsulation of functional blocks[1]. The paper presents both successful and unsuccessful endeavors to inform replications of this work and highlights avenues for feature tuning and scalability of the autograding model to larger, more diverse classrooms[1].

Despite the need for better feature embedding, the authors also explore options for clustering, starting with the most popular choice, K-means clustering[1]. This work integrates the academic community into the research and development of the optimal feature embedding of Snap! programs[1]. The paper was presented at the KDD '23 conference, held from August 06–10, 2023, in Long Beach, CA[1].

Source: Conversation with Copilot, 5/24/2024

(1) Extracting Features for Computational Thinking from Block-Based Code. https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/damle2023extracting.pdf.

(2) KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge .... https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/toc.html.

(3) Sensor Placement for Learning on Networks - kdd.org. https://www.kdd.org/kdd2023/wp-content/uploads/2023/08/burudgunte2023sensor.pdf.

(4) undefined. https://doi.org/X.