

Summer 2019

Finding Predictors of Success in Novice Programmers' Editing and Testing Behaviors

SaraJane Griffiths
sgriffiths@pugetsound.edu

Follow this and additional works at: https://soundideas.pugetsound.edu/summer_research

Recommended Citation

Griffiths, SaraJane, "Finding Predictors of Success in Novice Programmers' Editing and Testing Behaviors" (2019). *Summer Research*. 360.
https://soundideas.pugetsound.edu/summer_research/360

This Article is brought to you for free and open access by Sound Ideas. It has been accepted for inclusion in Summer Research by an authorized administrator of Sound Ideas. For more information, please contact soundideas@pugetsound.edu.

Background

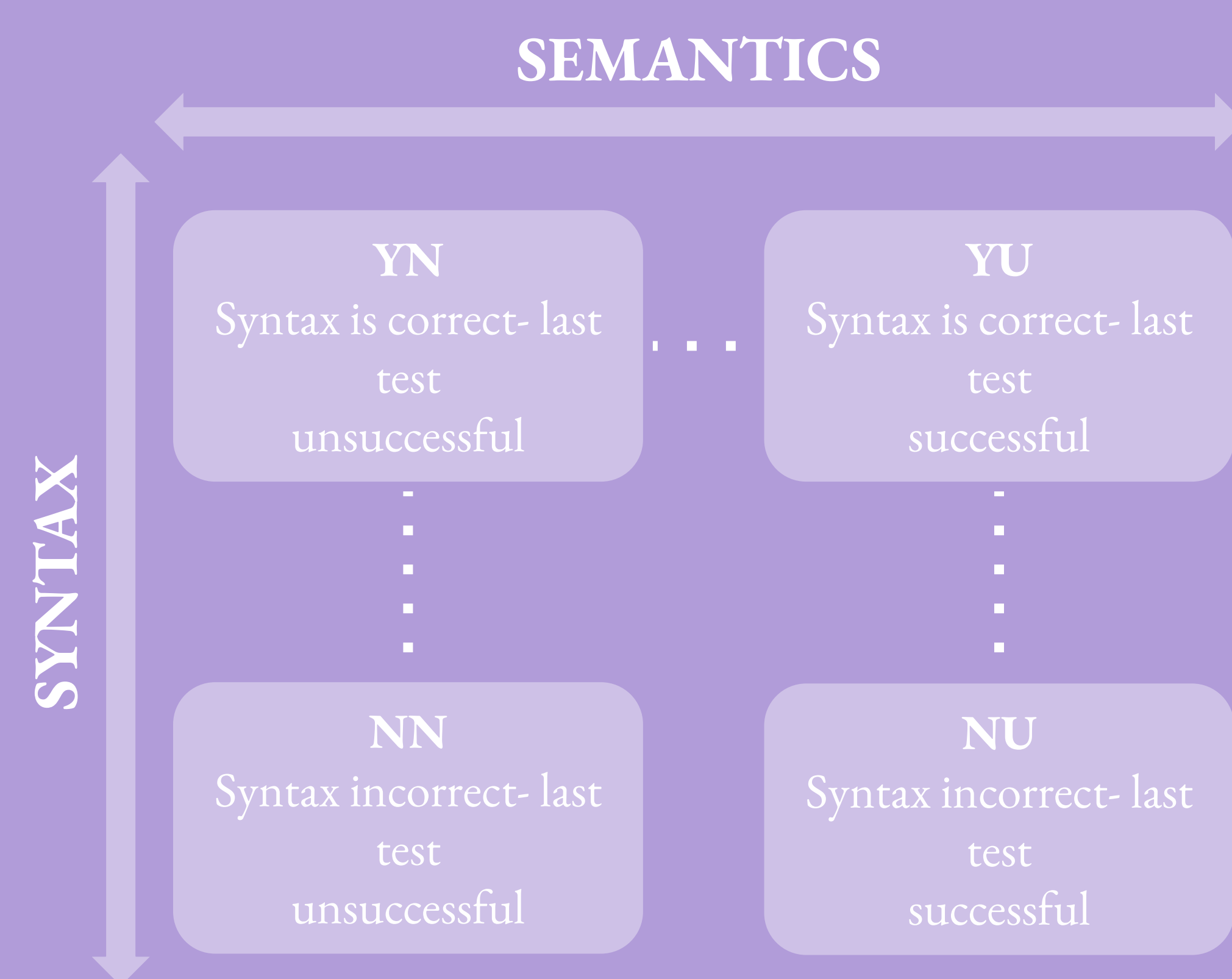
- The Normalized Programming State Model is promising for predicting the success (grades) of intermediate-level programmers.
- Previous research at the University of Puget Sound, by Ayse Hunt and Professor Brad Richards, found that NPSM was not likely to work with novice programmers.
- They used a large dataset that was anonymized, and there were no grades for the traces.

Research Questions

1. How bad is the Normalized Programming State Model in predicting grades for novice programmers?
2. Can NPSM be adapted to better predict novice programmers' success?

The Model

- Tracks the correctness of student programs as they are being developed.
- Records percentage of time spent in the states below.



The Grading Tool

- Finishing analysis on the previous research required thousands of student programs to be automatically graded.
- We built automated software tests for each program that assigned the program a grade.
- Student code differed in many ways, including method names, class names, variable types, and variable names.
- We developed a tool that automatically rewrote student programs to make them more uniform.

Finding Predictors of Success in Novice Programmers' Testing and Editing Behaviors

SaraJane Griffiths, Professor Brad Richards
sgriffiths@pugetsound.edu,
brichards@pugetsound.edu

Do Techniques For Predicting Intermediate Programmers' Success Work For Novices? Probably Not!

The Altered Model

- NPSM judges semantic correctness by whether a student's program crashes, but novice code is often not complex enough to crash, even if it is incorrect.
- We hypothesized that if a student tested a piece of their program, then returned to editing that same piece, the program probably didn't work as intended.
- We developed an altered NPSM model that judges semantic correctness based on that behavior pattern.

Results

- We used regression to produce grade prediction formulas for the models, based on the percent of time spent in states.
- Compared the predicted grades to the grades produced by the automatic grading tool.
- Preliminary results show that the altered model performs slightly better.
- Neither model did very well at predicting novice grades.

Future Work

- Look into more complex/longer exercises within the repository of novice programs
- Examine subsets of the current data set for trends and/or patterns.
- Look at the accuracy of the prediction formulas in predicting whether students are passing or failing

References

- Adam S. Carter, Christopher D. Hundhausen, and Olusola Adesope. 2017. Blending Measures of Programming and Social Behavior into Predictive Models of Student Achievement in Early Computing Courses. *ACM Trans. Comput. Educ.* 17, 3, Article 12 (August 2017), 20 pages. DOI: <https://doi.org/10.1145/3120259>
- Adam S. Carter, Christopher D. Hundhausen, and Olusola Adesope. 2015. The Normalized Programming State Model: Predicting Student Performance in Computing Courses Based on Programming Behavior. In *Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15)*. ACM, New York, NY, USA, 141-150. DOI: <https://doi.org/10.1145/2787622.2787710>
- Brad Richards and Ayse Hunt. 2018. Investigating the Applicability of the Normalized Programming State Model to BlueJ Programmers. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling '18)*. ACM, New York, NY, USA, Article 10, 10 pages. DOI: <https://doi.org/10.1145/3279720.3279731>
- Marzieh Ahmadzadeh, Dave Elliman, and Colin Higgins. 2005. An analysis of patterns of debugging among novice computer science students. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITICSE '05)*. ACM, New York, NY, USA, 84-88. DOI: <https://doi.org/10.1145/1067445.1067472>
- M. C. Jadud and B. Dorn. Aggregate compilation behavior: Findings and implications from 27,698 users. In *Proceedings of the eleventh annual International Conference on International Computing*.

I would like to express my appreciation to the McCormick Research Grant at the University of Puget Sound for funding this research and to Professor Richards for his support and mentorship.