Coding in the Liberal Arts through Natural Language Processing and Machine Learning

Ursula Wolz, Jennifer Wilson

Eugene Lang School of Liberal Arts, The New School, New York, NY, USA
wolzu@newschool.edu, wilsonj@newschool.edu

Abstract

An initiative recently established at our institution is creating new opportunities for students to deepen their understanding of code and computational thinking, and to embrace questions of access, equity and social justice. In this short paper we report on two contextualized computing courses in this initiative that introduce coding and computational thinking through contextualizing two subfields of AI: Natural Language Processing and Machine Learning. The goal was two-fold: to help students gain foundational computational skills to further their own creative and critical practices; and more broadly, to help them develop better-informed critiques of the use of algorithmic systems, especially AI technology.

Coding in the Liberal Arts

At Eugene Lang School of Liberal Arts, an initiative that led to a new minor in Code as a Liberal Art provides opportunities for students to explore computation and embrace questions of access, equity and social justice. The goal was two-fold: to help students gain skills in coding and computational thinking to further their own creative and critical practices; and to help them develop better-informed critiques of the use of algorithmic systems in social and economic interactions and decision-making. Thus, the initiative is concerned with developing coding literacy, but also addresses a new form of civic education.

With strengths in the Social Sciences, Media studies, and the Arts and Writing, Lang has no stand-alone mathematics department and no computer science. This provides a unique perspective for faculty because the courses are not for ‘non-majors,’ nor do they contribute to a major in a liberal arts setting (Baldwin and Brady, 2010).

A set of common learning outcomes for the courses in the a Liberal Art minor articulate goals of the program within a liberal arts framework. These include the following: Students learn to use computation as a tool to analyze, communicate, create, and learn. They develop collaborative skills utilizing computational and algorithmic thinking. They also gain an understanding of the historical and social factors leading to the increasing presence of computational systems, and develop an ethical framework as they work through the social and political implications of, and embedded within computing technologies. They learn to appreciate the challenges of equity and access to computing technologies as well as how technologies can reinforce inequalities. They learn to thinking critically about computer-mediated interaction, understanding its limits from philosophical, logical, mathematical and policy perspectives. Finally, each course attempts to help them understand the intrinsic relationship between the physical world, analog environments and digital experiences.

Courses within the minor are therefore contextualized computing courses as articulated by Xu, Wolz and Greenberg (2018) who suggest that foundational computer science courses should balance a partnership between disciplines, and provide exposure to computer science that is *experiential.* Students must code to learn about the context, and they learn to code through problems provided by the context. The courses described here provide context drawn not only from Artificial Intelligence but also from Linguistics, Poetry, Psychology, Philosophy and Learning Science.

Liberal Arts AI

Artificial Intelligence, and more specifically Natural Language Processing (NLP) and Machine Learning (ML) are having a profound influence on daily life. The ubiquitous influence of recognition algorithms, whether for voice, text, or faces, as well as search engines and recommender systems increasingly impacts human decision-making at all levels. Citizens have a responsibility to educate themselves regarding the effects of these processes, to understand how they work and how AI grew out of well-established fields. As part of liberal learning, the question arises as to what and how to teach AI. A 2016 survey of AI educators and practitioners (Wollowski et al. 2016) suggests that core concepts in knowledge representation and reasoning are critical to both academics and practitioners. While academics also listed games and puzzles, practitioners listed applications. As suggested by Wollowski et al., games are an ideal vehicle for instruction, because they simplify the inherent complexity of real-world systems.

Two courses were offered at Lang, 'Coding Natural Language' (Fall 2018) and 'Do Machines Learn?' (Spring 2020). Enrolled students were undergraduates (at all levels) from Lang as well as Parsons School of Design with interests ranging from Fashion to Mathematics. There were no prerequisites; about 1/3 had some prior coding. Neither course was intended as a robust first course in AI. The goal was to provide a critical overview of these disciplines through the experience of coding.

Coding Natural Language

'Coding Natural Language' introduced traditional knowledge-based AI techniques. The course covered material that met the Code as a Liberal Art objectives summarized above. Students wrote code in Python developing skills using a method we call '*DAHLIA*' to define **D**ata, express **A**lgorithms using **H**euristics and **L**ogic, explore **I**nterface design and model **A**bstractions. They practiced agile software development and scientific method. They used and explained NLP concepts including Phonology, Morphology, Syntax, and Semantics. Reading in computational linguistics and philosophy of mind showed the power and limitations of formal and natural language.

Each student completed a web-based journal, a series of small projects, and a large self-designed project. The small projects explored template systems via a Madlibs assignment, generative and kinetic poetry to understand formal grammars and semantics (with a Poet guest lecturer), text-based game play to explore meaningful interaction, disambiguation (via NLTK), summarization (via an NL Generation researcher who gave a guest lecturer), and stochastic methods in natural language translation.

Do Machines Learn?

'Do Machines Learn' focused on the power of human learning and the limitations of ML. The core objectives of the minor were met here too. Students constructed computational solutions to a problem using the *DAHLIA* model in JavaScript to practice core concepts in programming fundamentals. A scaffolding collaborative project explored AI including search, planning, logic, neural nets, robotics and machine learning. Students did research projects on perspectives of human learning, ranging from neural science to psychiatry to learning differences. To further meet core objectives they engaged in discussions about the credibility, ethics, and social impact of search, agent theory and machine learning systems. A final exam question asked them to defend or dispute the proposition that machines 'learn.' A plurality asserted that ML is a form of behaviorism, but doesn't quite meet other types of natural learning.

The course emphasized collaboration through a single major project using agile software development and scientific method. Students built a player-less game (no human players; programmed avatars were dropped into the world model.) Degrees of agency were programmed into avatars. Students collaborated on a world knowledge representation. Individual avatars began as reactive agents, then employed classic A* search, and then adversarial search to find resources and avoid danger. Well-formed heuristics proved powerful to effect efficient survival. An interface to the ML system tensorflow.org was able to identify objects in the world, but integrating the network code into individual agents proved too complicated for novice coders.

Outcomes and Next Steps

Outcomes from both courses were positive: the lowest grade received in each was a 'B' implying that learning goals were met. Surveys administered by Lang in Fall 2018 for the 'Coding Natural Language' course showed great enthusiasm for the course. No surveys were administered in Spring 2019. Informal interviews in both semesters corroborated the formal feedback. Analysis of student work and feedback suggested that 'Do Machines Learn' could be improved by offering it as a second level course to focus more directly on data structures, algorithms, and network communication to access ML online systems. A second level version will be offered in Spring 2020. More systematic data gathering will take place in future iterations of all Code as a Liberal Arts courses. Course materials may be obtained by contacting the authors.

References

Baldwin, M. and Brady A. Guest Editors: Computer Science in the Liberal Arts 2010. Transactions on Computing Education, 10(1):1:1-1:5

Wollowski, M.; Selkowitz, R.; Brown, L. E., Goel, A.; Luger.G, Marshall, J.; Neel. A,; Neller, T.; Norvig, P. 2016. A Survey of Current Practice and Teaching of AI. In *Proceedings of the Sixth Symposium on Educational Advances in Artificial Intelligence*, 4119-4124. Menlo Part, Calif: AAAI Press.

Xu, D.; Wolz, U; Greenburg, I. 2018 Updating Introductory Computer Science with Creative Computation. *In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18).*, 167-172. ACM, New York, NY, USA.