“Flipping” for Journalism Tech Education

Susan E. McGregor
Tow Center for Digital Journalism
Columbia Journalism School
sem2196@columbia.edu

ABSTRACT

In recent years, the ability to upload and distribute video through free online platforms like YouTube, Vimeo and others has made it feasible for educators to fully leverage the teaching and learning possibilities of user-paced video for instruction. Such videos make possible a “flipped classroom” teaching approach, wherein students watch screencasts of the class material *prior* to the in-person class session. Combined with direct discussion and collaborative, peer-learning techniques, this model supports effective and scalable programming and technology education for journalism and communications students. In this case, the effectiveness of the method is enhanced by framing the foundational concepts of programming in terms of a journalistic story structure.

INTRODUCTION

Tracking changes in professional reporting and publication practices, the demand for effective data- and technology-related education for journalists and journalism students has exploded in recent years. Attendance at the National Institute for Computer-Assisted Reporting (NICAR) conference has multiplied in the last ten years, with data analysis and basic programming among the skills “every aspiring journalist should learn.” This sentiment has clearly been taken to heart in journalism schools, where new initiatives ranging from certificate programs to MOOCs (e.g. the Lede Program, Stanford’s “refresh” M.A. program, The Knight Center for Journalism in the Americas’ “Data-Driven Journalism: The Basics”) have been developed to address the skyrocketing demand for these essential skills.

Given the incredible demand for data- and engineering-related skills and the variety of programs being developed to deliver them, now is an important moment to evaluate the effectiveness of specific educational interventions. In this paper, I offer a case study of one such approach: the use of a contextually-grounded, “flipped classroom” approach to teaching data and programming skills to journalism students.

BACKGROUND AND RELATED WORK

Alongside a considerable body of literature in education and instructional design addressing the importance of including realistic, contextually-relevant problems in classroom activities, recent work in education research has explored the specific pedagogical effects of the so-called “flipped classroom” model made possible by the widespread availability of inexpensive or free online-video platforms. The following sections provide a brief overview of existing work in these areas.

Making the Abstract Concrete

One of the major challenges of STEM-related education is the relative abstractness of many of the concepts involved. This is especially true in the case computer programming, where – unlike in the physical sciences like biology and physics, for example - there is no definitive analog between the constructs and processes of a computer’s operation and any visible process in the physical world.

Because of this, topics like computer programming can suffer from an especially high barrier to entry. This is true particularly for adult learners, who do not absorb information as quickly as their younger counterparts, and who may need to revise existing mental models about computer systems in order to engage successfully with programming tasks. Related to this, adult learners may need to reframe their own self-concept with respect to computers and computing, which can produce significant anxiety. Successfully addressing these concerns is an important part of the learning process for these students.

One method for mitigating the learning challenges posed by substantially abstract subject matter is to intentionally ground instructional materials in relevant, concrete and familiar tasks (Felder and Silverman). Therefore, rather than focusing on generalized or atomized programming or data-analysis skills, the methods discussed below intentionally connect each new skill to a concrete, achievable task that is situated in the context of a journalistic story structure or reporting practice (Lave & Wenger).

“Flipping” the Classroom

In recent years a number of studies have been conducted on the various effects of a “flipped” classroom approach for teaching science and engineering topics. Generally speaking, a “flipped” classroom approach has two primary features: first, students are required to engage in initial learning activities (typically watching video presentations or doing readings) in *advance* of related classroom activities; second, classroom activities themselves are generally more interactive and problem-based. According to work by both Bishop and Verleger and Herreid and Schiller, the application of “flipped” classroom techniques to STEM subjects tends to both improve student perception of the course as well as their performance, though some student strongly dislike what may be an unfamiliar approach.

METHODS

The following pedagogical approach was iteratively refined over four years of teach two distinct courses at Columbia University's Graduate School of Journalism. The first course is a seven-week introductory course in data journalism that was offered in the latter half of the fall semester. The second course is a fourteen-week spring seminar on information visualization. All courses were offered between the fall of 2012 and the spring of 2015. The number of students per section ranged from eleven to seventeen. No prior experience in data analysis, visualization, or programming was required.

Students

The background and prior experience of students in these courses varied considerably. Student ages ranged from early twenties to mid-forties, with some students arriving with substantial experience in data analysis and/or programming. Others had no prior exposure to these practices and in some cases had minimal computing skills.

Instructional Design

The instructional design methods used in these courses draw heavily on constructivist educational theories and methods, and incorporate strategies for grounding, framing and signaling of content for learners in addition to support for self-paced learning, direct and indirect learning, and prompted reflection for concept retention and transfer. For clarity, elements of the instructional design are presented here roughly in the order that students would encounter them during each course.

Addressing Stereotypes and Self-Concept

Stereotype issues around programming and technology education can be particularly difficult for humanities-focused students who may feel that they are "not math or science people." Indeed, for many years journalistic culture exhibited unabashed disdain for mathematical and/or technical proficiency, which, though it has given way to conversations about the importance of data and programming literacy, has not necessarily articulated a clear way forward for many students whose comfort area is still the written word.

Because of this, directly addressing stereotypes around what "kinds" of people do data and programming is the *very first* classroom discussion conducted in both the data journalism and the information visualization course. Using the introduction to Douglas Rushkoff's *Program or Be Programmed* as a starting point, students are encouraged to actively share their impressions of and anxieties about what programming requires. As an instructor, this offers me the opportunity to point out to students that many of are already expert in (or feel themselves equal to) incredibly complex tasks (such as Rushkoff's chosen analogy of driving), but that their sense of reduced ability is likely fueled by the *invisibility* of most programming work, as well as by media-driven images of the types of people who program.

The effect of these initial exercises is two-fold. First, the sharing of impressions and anxieties in a group has been shown to help relieve them (Bielaczyc & Collins, 269-292). Second, by highlighting to students that they are *already proficient* in a range of complex tasks, they experience a positive pressure to engage the difficult material of learning to do data analysis and programming (Driscoll, 377).

Structure of Video Content and Classroom Activities

In keeping with a typical "flipped" classroom structure, each week students are provided with links to a series of videos containing the upcoming week's lab activities. While locating quality video material has proven a challenge in some flipped-classroom scenarios (Bishop and Verleger), this obstacle is avoided here through the creation of new videos each week throughout the term. While somewhat more labor-intensive for the instructor, this approach offers many advantages over externally-produced instructional materials. First of all, their contents are guaranteed to be timely and one-hundred percent relevant to the specific course being taught. Beyond this, however, there are qualitative features of these "homemade" videos that have specific pedagogical advantages over pre-produced segments.

For example, the "screencast" model of video production, in which the instructor narrates learning activities in real-time while carrying them out, means that the style of videos is intentionally unpolished and casual, making the material seem more approachable to learners (Shank, Berman & Macpherson, 161-182). Likewise, it is inevitable that a video created with minimal rehearsal will include some mistakes in procedure, and documenting and modeling error-recovery is a form of indirect instruction that is especially important in ill-defined and frequently-changing subject areas like programming (Ruutman & Kipper). Another "side effect" of the low production cost of such screencasts is that the freely available software used imposes a fifteen-minute time limit on a each video, meaning that most week's playlists are made up of three to four individual videos. This naturally requires that the material for each week be segmented and framed appropriately to orient students to the contents of each individual video. This results in better framing and signaling of upcoming content, which generally results in better learning and content retention (Payer & Pilegard, 316-344).

Finally, the very fact that the videos are available online means that students can benefit from all of the well-documented learning advantages of self-paced video instruction, viewing and reviewing them as needed (Scheiter, 487-512).

Code Commenting and Prompted Reflection for Learning

Consistent with the typical "flipped" classroom structure, students often complete the majority of each week's programming or data analysis assignment during the class meeting. The complete assignment for each week, however, always has specific features designed to enhance concept retention and transfer.

The first of these features is a primary grading focus on the quality, clarity, and thoroughness of commenting throughout each programming assignment. In general, students are expected to use comments to describe, in meaningful English sentences, what each block (or sometimes line) of code within their program is doing. The reason for this is simple: a functioning program is not a clear indicator of conceptual understanding. By emphasizing the quality of students' comments as the primary evaluation mechanism for programming assignments, it helps reduce student anxiety (by placing a premium on the part of the task with which they are probably most comfortable, i.e. expressing themselves in English, rather than code). More importantly, the English-language descriptions of the code activity illustrate much more effectively both the level of students' conceptual understanding of the programming concepts they're using, and, where they are inaccurate, an indication of where and how their reasoning may have gone awry. This makes it much easier, on a weekly basis, to provide meaningful and effective feedback to students.

Similarly, as part of their final projects, students are required to submit a reflection document in which they describe how they went about reporting, writing and building the materials for their piece. As with the commenting on individual programming assignments, this "prompted reflection" helps solidify in students' minds what was successful and unsuccessful about their approaches, which is an important component of the learning process. In general, this process of reflecting on and articulating one's thought processes and understandings has been shown to improve concept-reinforcement, retention, and skills-transfer (Wiley & Chi, 413-432).

Content Design

Finally, in keeping with the idea of rooting abstract learning material in concrete tasks and ideas, the design of many of the learning materials used in these courses draws explicit parallels between the processes and products of journalism and those of computer programming.

For example, many fundamental concepts around the organization of a computer program can be usefully compared to the structure of a journalistic article written in English (the language of instruction at Columbia Journalism School). In both cases, the "reader" starts at the top, and reads left to right, and top to bottom. Variable declaration in the program is likened to introducing a new character in a story; in both cases, one can't simply refer to a person (or variable) that hasn't already discussed "further up" in the "text." Similarly, complex JavaScript data types, like objects, are introduced by translating information about the subject of a real news story to JavaScript object notation, where the "person" variable is declared with introductory details, and then that object is updated as the story is read, through the addition or modification of named properties (e.g. name, age, height, occupation, siblings &c). By drawing these concrete parallels between the grammar of programming and the structure of news stories, journalism students are able to more quickly and effectively anchor their understanding of these fundamental concepts.

RESULTS

While the results of this work are currently anecdotal, the observed results of this instructional design approach to teaching data analysis and programming to journalism students is largely in keeping with prior results in "flipped" classroom courses.

In the course of four years, none of the more than one hundred students to take one of these courses has failed to satisfactorily complete the required individual programming assignments, of which there are between five and ten in each course. Likewise, the most positive comments about the utility of the online videos has come unsolicited from some of the students who struggled most with the technical material, suggesting that this type of instruction is particularly helpful for less experienced or more intimidated learners. At the same time, experienced students have favorably contrasted the interactivity of the classroom activities with prior coursework in more traditional programming classes, with one even stating, "If it had been this much fun, I probably would have stayed in computer science."

DISCUSSION AND CONCLUSION

The rising demand for data and technology-related education in journalism means that the successful teaching approaches need not only to support effective and meaningful learning outcomes, but must also scale in such a way that these courses can reach more than a handful of students. An essential component of the approach outlined above is the reduced burden it places on individualized instruction both in the classroom and in office hours. The minimal production value of the videos used not only supports the actual learning goals of the course, but it minimizes the time and effort required to produce them: in essence, the instruction has only to design and prepare each week's lesson a few days in advance in order to generate the videos. This approach also has the added advantage of ensuring sufficient preparation for each lab meeting, and an opportunity to test existing approaches for currency, which is especially important as the technologies used in these fields changes rapidly.

Likewise, while smaller classes are preferably for the interactive classroom activities, the videos themselves can be used by an unlimited number of learners and watched as many times as necessary, reducing the burden on instructors' limited office hours and email availability. Supplemented with an effective learning management system (LMS) that allows students to centrally post questions and answer them for one another, a great deal of learning can be supported by a single instructor. All of these features suggest that a "flipped," grounded, collaborative approach to data analysis and programming instruction in journalism offers an opportunity to effective scale this kind of instruction across learning institutions and communities.

REFERENCES

Bielaczyc, Katherine and Collins, Allan. "Learning Communities in Classrooms: A Reconceptualization of Educational Practice" , Ed. Charles M. Reigeluth. *Instructional-Design Theories And Models*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983. *eBook Collection (EBSCOhost)*. Web. 19 Mar. 2015.

Bishop, J. & Verleger, M. "The Flipped Classroom: A Survey of Research," *120th ASEE Annual Conference & Exposition*, 23-26 June, 2013. http://www.studiesuccesho.nl/wp-content/uploads/2014/04/flipped-classroom-artikel.pdf

Driscoll, Marcy P. *Psychology of Learning for Instruction*. 3rd ed. Boston: Pearson, 2005.

Felder, Richard M. and Silverman, Linda K. "Learning and Teaching Styles in Engineering Education," *Engineering Education*, 78(7), 674-681. 1988.

Herreid, C. & Schiller, N. "Case Studies and the Flipped Classroom," *Journal of College Science Teaching*. 42 (5), 2013. p. 62-66.

Lave, J., & Wenger, E. (1990). Situated Learning: Legitimate Periperal Participation. Cambridge, UK: Cambridge University Press.

Mayer, Richard E. and Pilegard, Celeste. "Principles for Managing Essential Processing in Multimedia Learning: Segmenting, Pre-training, and Modality Principles," Mayer, Richard E., ed. *The Cambridge Handbook of Multimedia Learning*. 2nd ed. Cambridge: Cambridge University Press, 2014. *Cambridge Books Online.* Web. 19 March 2015. **http://dx.doi.org.ezproxy.cul.columbia.edu/10.1017/CBO9781139547369**

Ruutmann, T.; Kipper, H., "Teaching strategies for direct and indirect instruction in teaching engineering," *14th International Conference on Interactive Collaborative Learning (ICL), 2011*, vol., no., pp.107,114, 21-23 Sept. 2011
doi: 10.1109/ICL.2011.6059556.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6059556&isnumber=6059533

Scheiter, Katharina. "The Learner Control Principle in Multimedia Learning" Mayer, Richard E., ed. *The Cambridge Handbook of Multimedia Learning*. 2nd ed. Cambridge: Cambridge University Press, 2014. *Cambridge Books Online.* Web. 19 March 2015. **http://dx.doi.org.ezproxy.cul.columbia.edu/10.1017/CBO9781139547369**

Shank, Roger C., Berman, Tamara R., Macpherson, Kimberli A. "Learning by Doing," Ed. Charles M. Reigeluth. *Instructional-Design Theories And Models*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983. *eBook Collection (EBSCOhost)*. Web. 19 Mar. 2015.

Wylie, Ruth and Chi, Michlene T. H. "The Self-Explanation Principle in Multimedia Learning" Mayer, Richard E., ed. *The Cambridge Handbook of Multimedia Learning*. 2nd ed. Cambridge: Cambridge University Press, 2014. *Cambridge Books Online.* Web. 19 March 2015. **http://dx.doi.org.ezproxy.cul.columbia.edu/10.1017/CBO9781139547369**