Tutorons - Code Explanation and Demostration Systems Review

Jiawei Yuan
University of Illinois at Urbana-Champaign

1 Introduction

The goal of this paper is to review existing code explanation and demostration systems to help programmers leverage web documentation to solve programming problems. First, we introduce the Tutorons[1] systems to analyze how it utilize web data to generate micro-explanations of codes. Second, we give the overview of some existing explanation and demonstration systems to compare them with the Tutorons system.

2 Tutorons Overview

A Tutoron is a routine on a web server with language-specific rules for detecting, parsing and explaining source code written on a web page. It can automatically find code segment on websites to generate simple explanations or demonstrations which is called micro-explanations. Those explanations are context-relevant, and the specific flags or usage with in the domain are demonstrated with natural language and examples.

To implement these functions, there are three stages to follow.

In the first stage called detection, a Tutoron extracts explainable regions from a HTML document using the language's lexicon and/or syntax. This consists of four steps: First, it extracts code blocks from pages. Second, it divides code blocks into candidate explainable regions. Third, it checks whether those regions follow the grammarl. Last, it filters candidates to reduce false positives.

The second stage is called parsing, which is similar to the parsing in compiling. It can parse code snippets into some data structure to get ready for explanation. There are two methods, the first one is to introduce hooks into existing parsers to extract the results of parsing. And the second one is to develop a custom parser for certain languages whose subset is supported.

The final stage is called explanation. The Tutoron traverses the data structure built in the second stage to generate explanations and demonstrations of the code. There are three major types that the Tutoron can generate micro-explanations for, CSS selectors, regular expressions, and wget.

[1] Head A, Appachu C, Hearst M A, et al. Tutorons: Generating context-relevant, on-demand explanations and demonstrations of online code[C]//2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). IEEE, 2015: 3-12.

3 Comparisons

The routine of the Tutoron contains two themes: automatic explanation and demonstration of code.

Webcrystal[2] is the most closely related to the Tutoron in the theme of automatic explanation of code. It is a web development tool that helps users understand how a web page is built. It contributes novel interaction techniques that let user quickly access HTML and CSS information by selecting questions regarding how a selected description and a customized code snippet that can be copied-and-pasted to recreate tthe desired properties. Moreover, it supports combining styles and structures from multiple elements into the generated code snippet, and provides visualizations on the web page itself to explain layout relationships.

Similar to Webcrystal, the Tutoron also generate human-readable representations of code to help programmers reuse and learn from online examples, and it also aims at HTML and CSS information on web pages. However, Webcrystal focuses on demonstrating overall and large-scale HTML/CSS elements, and on the contrary, the Tutoron puts it efforts on describing short, embedded languages like regular expressions and Unix commands instead of HTML, and also develops guidelines for generating effective explanations for these languages.

As for the automatic demonstration, the Tutoron draws inspiration broadly from various visual tools which aid in programming instruction.

Sorva[3] presents a pedagogical technique called visual program simulation which involves the learner in the interactive simulations in which the learner takes on the role of the computer as the executor of a program. Python Tutor[4] is a web-based program visualization tool, with which students can write python programs, debug them without any plugins and share the program visualization. Recently, Ou et al.[5] produced visualizations of pointer-based data structures in the heap. And this work can offer interactive tools to graphically interact with the data and program execution process. D'Antoni et al.[6] created counterexamples of system behaviors to provide helpful feedback for computer science students doing automata design problems, which could help student increase perseverance and ability in solving problems.

The tutoron gained its insight from these relevant work in visualization and example generation in the human computer interaction field to produce demonstrations of the code snippets. Besides that, it can produce guides for generating those explanations of code on the online website, which is quite unique and

[2] Chang K S P, Myers B A. WebCrystal: understanding and reusing examples in web authoring[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2012: 3205-3214.

[3] Sorva J. Visual program simulation in introductory programming education[M]. Aalto University, 2012.

[4] Guo P J. Online python tutor: embeddable web-based program visualization for cs education[C]//Proceeding of the 44th ACM technical symposium on Computer science education. 2013: 579-584.

[5] Ou J, Vechev M, Hilliges O. An interactive system for data structure development[C]//Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. 2015: 3053-3062.

[6] D'antoni L, Kini D, Alur R, et al. How can automatic feedback help students construct automata?[J]. ACM Transactions on Computer-Human Interaction (TOCHI), 2015, 22(2): 1-24.

helpful than predecessors.

4 Conclusion

Overall speaking, the Tutoron produce explanations for programmers to get a better understanding of certain code snippets. Moreover, comparing to the related work, it focuses more on short code segments and can support many languages. And it can produce manuals on how to generate those explanations, which can help people utilize this tool in order to build a more comprehensible website.