Chapter 20
Play–by–Play Learning for Textual User Interfaces

Nate Blaylock
Florida Institute for Human and Machine Cognition, USA

William de Beaumont
Florida Institute for Human and Machine Cognition, USA

Lucian Galescu
Florida Institute for Human and Machine Cognition, USA

Hyuckchul Jung
Florida Institute for Human and Machine Cognition, USA

James Allen
University of Rochester, USA

George Ferguson
University of Rochester, USA

Mary Swift
University of Rochester, USA

ABSTRACT

This chapter describes a dialog system for task learning and its application to textual user interfaces. Our system, PLOW, uses observation of user demonstration, together with the user’s play-by-play description of that demonstration, to learn complex tasks. We describe some preliminary experiments which show that this technique may make it possible for users without any programming experience to create tasks via natural language.

INTRODUCTION

Our daily activities typically involve the execution of a series of tasks, and we envision personal assistant agents that can help by performing many of these tasks on our behalf. However, for computers to perform tasks for us, they must be “taught” how to do those tasks. One way to do this is traditional programming: an experienced programmer specifies the task in a computer language. This is the current method for teaching computers how to do things for us, and there are a lot of useful programs out there. Unfortunately, several factors severely limit the effectiveness of this traditional method: first, the ability to program resides with a very small portion of the population; second, creating a program can be expensive (in both money and time); and lastly, there are a great number of tasks that are specific to an individual or a small group. The combination of these factors means that a
great number of tasks do not get implemented, not because they would not be useful, but because most people cannot program their own tasks and the economics makes it prohibitive for individuals or small groups to get tasks written for them.

Another possible method for learning tasks is by observation. Researchers have attempted to learn these models by having the computer observe the user’s actions when performing a task (Angros et al. 2002; Lau and Weld 1999; van Lent and Laird 2001; Faaborg and Lieberman 2006). However, these techniques require multiple examples of the same task, and the number of required training examples increases with the complexity of the task.

Previously, we presented work on the PLOW system (Jung et al. 2008; 2010), which is able to learn tasks on the web from only a handful of examples (often even with a single example) through observation accompanied by a natural language (NL) “play-by-play” description from the user. The play-by-play approach in NL enables our task learning system to build a task with high-level constructs that are not inferable from observed actions alone. The synergy between the information encoded in the user’s NL description and observed actions makes it possible to learn, from a simple sequence of actions, a complex task structure that reflects the user’s underlying intentions in the demonstration.

Intuitively, the approach taken by PLOW is that the user “shows” the computer how to do a task, much in the same way he would show another person. In this sense, when the user is teaching, PLOW is looking over the user’s shoulder, watching what is done on the screen, and also listening to the user’s description and explanation of the actions performed. As most humans have the capability to show other humans how to do something, our hope is that this type of interaction will allow any person, including those with little or no programming experience, to be able to easily and quickly teach the computer a task.

The Application Domain

The Composite Health Computer System (CHCS) is a textual user interface system used throughout the US Military Health System for booking patient appointments and other tasks. In CHCS, the keyboard is used to navigate through a complex web of menus and screens in order to book and cancel appointments, look up patient information, and perform other administrative tasks. A typical screenshot of the system is shown in Figure 1. In the top part of the figure, spacing is used to make visual tables. The bottom shows a mnemonics-based menu and the cursor awaiting user keyboard input.

In this chapter, we discuss the application of the PLOW system to textual user interfaces, specifically the CHCS system. In developing and evaluating the system, we were given access to the actual CHCS server in use at Naval Hospital Pensacola, although, to protect live data, testing was done on a fictitious database used for training staff on CHCS use. As part of the development process, we conducted interviews and observation sessions with several types of diverse system users including nurses, clerks in specialist departments, and operators in the hospital call center. The call center was chosen as the focal point for our evaluations, as the high volume of calls and tasks performed lent themselves to testing.

Challenges

In this paper, we discuss our work on porting the PLOW system to the appointment booking domain in CHCS. In moving to this domain, we addressed the following challenges:

- **Learning Collaborative Tasks:** While the PLOW system has been successfully applied to various tasks on the Web (e.g., finding restaurants/flights/etc.), the tasks performed by the system were mostly performed in a batch style in that, given a task
Related Content

Departing the Ontology Layer Cake
www.igi-global.com/chapter/departing-the-ontology-layer-cake/133879?camid=4v1a

Comparison between Internal and External DSLs via RubyTL and Gra2MoL
www.igi-global.com/chapter/comparison-between-internal-and-external-dsls-via-rubytl-and-gra2mol/108753?camid=4v1a

Audio and Speech Watermarking and Quality Evaluation
www.igi-global.com/chapter/audio-speech-watermarking-quality-evaluation/4686?camid=4v1a

Robustness Against DA/AD Conversion: Concepts, Challenges, and Examples
www.igi-global.com/chapter/robustness-against-conversion/8335?camid=4v1a