NLP (Natural Language Processing) for NLP (Natural Language Programming)

Rada Mihalcea[1], Hugo Liu[2], and Henry Lieberman[2]

[1] Computer Science Department, University of North Texas
rada@cs.unt.edu
[2] Media Arts and Sciences, Massachusetts Institute of Technology
{hugo, henry}@media.mit.edu

Abstract. Natural Language Processing holds great promise for making computer interfaces that are easier to use for people, since people will (hopefully) be able to talk to the computer in their own language, rather than learn a specialized language of computer commands. For programming, however, the necessity of a formal programming language for communicating with a computer has always been taken for granted. We would like to challenge this assumption. We believe that modern Natural Language Processing techniques can make possible the use of natural language to (at least partially) express programming ideas, thus drastically increasing the accessibility of programming to non-expert users. To demonstrate the feasibility of Natural Language Programming, this paper tackles what are perceived to be some of the hardest cases: steps and loops. We look at a corpus of English descriptions used as programming assignments, and develop some techniques for mapping linguistic constructs onto program structures, which we refer to as programmatic semantics.

1 Introduction

Natural Language Processing and Programming Languages are both established areas in the field of Computer Science, each of them with a long research tradition. Although they are both centered around a common theme – "languages" – over the years, there has been only little interaction (if any) between them[1]. This paper tries to address this gap by proposing a system that attempts to convert natural language text into computer programs. While we overview the features of a natural language programming system that attempts to tackle both the descriptive and procedural programming paradigms, in this paper we focus on the aspects related to procedural programming. Starting with an English text, we show how a natural language programming system can automatically identify steps, loops, and comments, and convert them into a program *skeleton* that can be used as a starting point for writing a computer program, expected to be particularly useful for those who begin learning how to program.

We start by overviewing the main features of a descriptive natural language programming system METAFOR introduced in recent related work [6]. We then describe in

[1] Here, the obvious use of programming languages for coding natural language processing systems is not considered as a "meaningful" interaction.

A. Gelbukh (Ed.): CICLing 2006, LNCS 3878, pp. 319–330, 2006.

detail the main components of a procedural programming system as introduced in this paper. We show how some of the most difficult aspects of procedural programming, namely steps and loops, can be handled effectively using techniques that map natural language onto program structures. We demonstrate the applicability of this approach on a set of programming assignments automatically mined from the Web.

2 Background

Early work in natural language programming was rather ambitious, targeting the generation of complete computer programs that would compile and run. For instance, the "NLC" prototype [1] aimed at creating a natural language interface for processing data stored in arrays and matrices, with the ability of handling low level operations such as the transformation of numbers into type declarations as e.g. `float-constant(2.0)`, or turning natural language statements like *add y1 to y2* into the programmatic expression `y1 + y2`. These first attempts triggered the criticism of the community [3], and eventually discouraged subsequent research on this topic.

More recently, however, researchers have started to look again at the problem of natural language programming, but this time with more realistic expectations, and with a different, much larger pool of resources (e.g. broad spectrum commonsense knowledge [9], the Web) and a suite of significantly advanced publicly available natural language processing tools.

For instance, Pane & Myers [8] conducted a series of studies with non-programming fifth grade users, and identified some of the programming models implied by the users' natural language descriptions. In a similar vein, Lieberman & Liu [5] have conducted a feasibility study and showed how a partial understanding of a text, coupled with a dialogue with the user, can help non-expert users make their intentions more precise when designing a computer program. Their study resulted in a system called METAFOR [6], [7], able to translate natural language statements into class descriptions with the associated objects and methods.

Another closely related area that received a fair bit of attention in recent years is the construction of natural language interfaces to databases, which allows users to query structured data using natural language questions. For instance, the system described in [4], or previous versions of it as described in [10], implements rules for mapping natural to "formal" languages using syntactic and semantic parsing of the input text. The system was successfully applied to the automatic translation of natural language text into RoboCup coach language [4], or into queries that can be posed against a database of U.S. geography or job announcements [10].

3 Descriptive Natural Language Programming

When storytellers speak fairy tales, they first describe the fantasy world – its characters, places, and situations – and then relate how events unfold in this world. Programming, resembling storytelling, can likewise be distinguished into the complementary tasks of *description* and *proceduralization*. While this paper tackles primarily the basics of