

Towards Programming in Everyday Language: A Case for Email Management

Toru Sugimoto, Noriko Ito, Shino Iwashita, and Michio Sugeno

RIKEN Brain Science Institute
2-1 Hirosawa, Wako, Saitama, 351-0198 Japan
{sugimoto, itoh, iwas, msgn}@brain.riken.jp

Abstract. In order to extend the application domain of natural language interfaces to more realistic tasks without the decrease of user’s performance, it is desirable for users to be able to specify their requests as coherent texts consisting of more than one sentence, in other words, to write a program in everyday language. In this paper, we present a processing model of a natural language interface that accepts such an input text. It consists of the text understanding process using a systemic functional linguistic resource called the Semiotic Base, and the mapping process from the structure of the input text to the structure of an output computer program. The algorithms explained in this paper have been fully implemented in our everyday language programming system that deals with personal email management tasks.

1 Introduction

Recent developments in natural language interfaces enable a wider range of people to access computing systems easily and friendly. Some systems only accept words or single sentences (i.e., commands) as inputs to the natural language interface, and other systems allow the users to converse with them interactively in either system/user initiative mode [1][3]. However, most of the existing systems interpret and process each sentence in the user inputs one by one.

In order to extend the application domain of natural language interfaces to more realistic and complex tasks without the decrease of user’s performance, users should be allowed to specify their complex requests and conditions as coherent texts consisting of more than one sentence, which are processed by the system in a unified way. Figure 1 shows an example of such an input text in a personal email management domain. It consists of three conditionals, which jointly specify a procedure for handling the user’s incoming emails. If we assume that “*the lab*” referred to in the second sentence is a part of “*RIKEN*”¹, and that “*lab*” and “*RIKEN*” are the names of separate mail folders, then naive interpretations of the first two sentences give us inconsistent requests for handling emails from people in the lab. So we need to modify naive interpretations to understand the user’s intention behind the input text as a whole. In general,

¹ “*RIKEN*” is the name of a research institute in Japan to which the authors belong.