Chapter 4

GENERATING UNDERSTANDABLE EXPLANATORY SENTENCES

Domenico Parisi
Donatella Ferrante

TABLE OF CONTENTS

1. Introduction
2. Explaining a third party's actions
3. Explaining events
4. Conclusion

1. INTRODUCTION

An important component of a language generation system is being able to produce explanatory sentences and texts. Operationally an explanatory sentence can be defined as a sentence which is an appropriate answer to a why-question:

(1) Q: Why did John go out?
   A: He wanted to buy a packet of cigarettes

Explanatory talk, of course, goes beyond producing appropriate answers to why-questions. What we may also want to call explanations are responses to requests for definition and clarification of terms, concepts, or ideas, or to questions about conclusions reached, methods followed, and so forth — all of which are usually not formulated as why-questions. However, for the purposes of the present paper, we will restrict our attention to giving explanations as answers to why-questions.

Even within this restricted subclass we must distinguish between explanations of human (and perhaps, animal) actions — e.g. John's going out — versus explanations of events — e.g. a house's falling down —, explanations of the system's own actions versus explanations of other people's actions, and so forth. The reason for making all these distinctions is that a proposed solution for one type of explanation may not be appropriate for other types. (For a discussion of some of these topics, see Wilks, 1977.) In the present paper we will deal separately with explanations of third party's actions (not the system's own actions; section 2) and explanations of events (section 3).

2. EXPLAINING A THIRD PARTY'S ACTIONS

Answering a why-question could be viewed as just a particular instance of answering wh-questions — i.e. questions about who, what, where, when, how, etc. — without posing special problems. Let us assume that a system's knowledge is represented as a network of nodes and arcs, with nodes representing the entities that the system has knowledge about and the network around each node representing the knowledge about the entity corresponding to the
node. Generating an answer to a wh-question can be conceived as a procedure composed of two parts. The first sub-procedure identifies a node (or nodes) in the knowledge network on the basis of the constraints provided by the question. For example, the question *Who is sleeping in the living room?* requires the system to identify one or more nodes on the basis of knowledge items such as (a) 'is a person', (b) 'is currently sleeping', (c) 'the sleeping takes place in the living room'. Once the node or nodes have been identified, the second sub-procedure directs the system to construct a linguistic expression for some additional knowledge that is found attached to the node or nodes, e.g. *John*.

From this point of view why-questions are not different from other kinds of wh-questions. The question *Why did John go out?* directs the system to identify a node in the network on the basis of the following constraints: (a) the node represents an event or fact causing another event or fact — this derives from the meaning of the word why; (b) the second event is that John goes out. Once this particular node has been found, the system selects some knowledge attached to the identified node and expresses this knowledge with the sentence *He wanted to buy a packet of cigarettes.*

This account of how why-questions are answered could be extended to cases where the knowledge about the identified node is not directly found attached to the node but must be inferred. In any case, the account might be judged as basically correct.

The problem appears to be more complex if we look at answering why-questions from a different perspective. Considered from the point of view of the person who has posed the question, answers to why-questions can be understandable or not understandable (we might also say: good or bad, or relevant or not relevant answers), with various degrees of understandability in between. Consider question (1) again:

(1) Why did John go out?

with the following list of potential answers:

(2) A. He wanted to buy a packet of cigarettes
    B. He wanted to go to Piazza Venezia
    C. He wanted to rub his nose
    D. He wanted to raise his left arm

For the system that generates any of these answers they may be all correct answers in the sense that each answer is based on the system's knowledge, whatever that knowledge may be. However to me, answer A is a good answer to question (1), but answers B-D are bad answers, with perhaps different degrees of badness among them. Now, if we want to construct a system that is able to generate answers to why-questions the system must be able to generate not any kind of answers but only good answers, that is, answers that prove to be understandable or relevant for the user.

In order to generate good answers to why-questions the system

- must possess a model of the user's knowledge store
- must be able to test a potential answer against the model and on the basis of this test conclude that the answer is a good or a bad one.

In other words, to generate good answers the system must try to understand its own potential answers using the user's knowledge store (or, more correctly, its model of the user's knowledge store). This implies that an important component of a sophisticated sentence generating system is a sentence understanding subsystem. (The role of user modelling in generating appropriate behavior is a widely studied topic in human-computer interaction. McKeown, Wish and Matthews (1985) have investigated how knowledge of the user's goals can help in generating explanations tailored to the user.)

We propose that to decide if a potential answer to a why-question is an answer which is understandable to the user the system must check if the answer satisfies three criteria. In order to explain these criteria we will have first to introduce some preliminary notions.