Generation of Natural Language Explanations of Rules in an Expert System*

María de los Ángeles Alonso-Lavernia,[1]
Argelio Víctor De-la-Cruz-Rivera,[2] and Grigori Sidorov [1]

[1] Center for Computing Research (CIC), National Polytechnic Institute (IPN),
Av. Juan de Dios Bátiz, Zacatenco, DF, 07738, Mexico
marial@uaeh.uaehred.mx, sidorov@cic.ipn.mx

[2] Center for Research on Technologies of Information and Systems (CITIS),
Autonomous University of Hidalgo State (UAEH), Mexico

Abstract. We present a domain-independent method for generation of natural language explanations of rules in expert systems. The method is based on explanatory rules written in a procedural formal language, which build the explanation from predefined natural language texts fragments. For better style, a specific text fragment is randomly selected from a group of synonymous expressions. We have implemented 16 groups of explanatory rules and 74 groups of explanatory texts containing about 200 text fragments.

1 Introduction

Expert systems are widely used to solve particular problems in a rather narrow area of expertise. They are based on knowledge obtained during interaction with human experts in the field, so they are also often referred to as *knowledge-based systems*.

One of important requirements for an expert system is the system's ability to explain its conclusions in a manner understandable to the user. The best form of presenting such an explanation is a text in natural language [5]. One approach to generation of explanations is to use as explanation the rules from the knowledge base that were fired during reasoning [6]. Another approach is writing special code that paraphrases the rules [8]. These approaches do not allow for description of the ideas behind the fired rules. An alternative is to use another knowledge database for generation of explanations [7]. This approach requires a double amount of work for constructing knowledge bases.

In this paper, we present a method that allows for representation of the ideas behind the rules, does not require any additional knowledge bases, and is domain-independent—i.e., it does not require reprogramming of an explanation system if the knowledge base is changed.

* This work was done under partial support of Mexican Government (CONACyT, SNI), Autonomous University of Hidalgo State, and National Polytechnic Institute, Mexico.

2 Generation of Explanations

Our system consists of:

- Explanatory rules written in a procedural formal language,
- Groups of predefined text fragments used by the rules to assemble sentences, and
- Textual representation of facts contained in the knowledge base (these facts are part of the knowledge-based production rules, thus, they are domain dependent).

Our target language was Spanish, though the same explanatory rules can be easily adapted for other languages. The main linguistic problem is agreement models that are specific to each language. Also, there are some other problems related to word order and restrictions on syntactic constructions [2]. All these language-dependent details can be easily taken into account when adapting our method to another language by modifying the code of the explanatory rules. We used the HAries [1], [4] programming language to implement the explanatory rules.

Explanatory rules are functions that perform specific operations for generation of the corresponding part of explanation, like verification of the number of arguments or analysis of the type of antecedent or consequent, etc. We use 16 groups of explanatory rules that is enough for construction of the whole system of explanations.

Texts fragments are inserted in the text slots of explanatory rules. They are classified as initial, medium, and final. The fragments are grouped together into sets of totally synonymous ones. For example, the group 1 contains "*Then this contributes*"; group 2 has "*the following condition*" and "*the following fact*", etc. The rules refer to such a group rather than to an individual fragment, and each time a random text is selected from the group to fill the slot, which makes the style of the explanations more vivid. We have 74 such groups of predefined texts with about 200 individual fragments. Some special text fragments are used for formatting in a way similar to HTML, representing text colors, line feeds, etc.

Textual representation of facts is taken from the knowledge base rules. Usually, knowledge-based systems contain knowledge in form of production rules. There are nine types of propositions that depend on the type of antecedent (left part of the rule) or consequent (right part of the rule)—whether it is a simple value or logical combination, and the presence of positive or negative intervals of weights associated with them. The system automatically generates the correct explanation depending on the type of rule.

The highest-level explanatory rule, which represents the whole explanatory system, is very simple (we omit changing of text attributes in the rule for simplicity, though we show them by underlining in the example below):

```
if (no antecedent in the rule) then
   print (text_1 + " " + text_2 + CONSEQUENT)
else
   print (ANTECENDENT + linefeed)
   if (a weight is given for antecedent) then
      print (text_3 + CONSEQUENT)
   else
      print (text_4)
   if (a weight is given for negation of antecedent) then
      print (linefeed + text_5 + linefeed + text_6 + CONSEQUENT_NEGATIVE)
```

The symbol + stands for concatenation. The expression *text_i* refers to a group of synonymous fragments; a specific expression is chosen randomly from the group. ANTECENDENT, CONSEQUENT, and CONSEQUENT_NEGATIVE are other explanatory rules, which in turn contain references to other such rules. Other explanatory rules, which are also rather short, are implemented in a similar manner.

We treat uncertainty by using corresponding natural language expressions for encoding of each uncertainty value o interval: for example, values from the interval 90% to 99% are expressed as *practically all*, *nearly in all cases*, etc.

Here is an example (in Spanish, see translation below) of an explanation generated using the knowledge base of oil production. Numbers in parenthesis stand for facts in the knowledge base.

Regla de producción generalizada "R1) 10 ⇒ 12 (-20 34)":
Si es establecido con peso absolutamente seguro el hecho siguiente:
    ({10} producción original ALTA)
Entonces esto contribuye con poca seguridad negativa (-20%)
al conocimiento sobre el hecho siguiente:
    ({12} Se pronostica el Efecto POSITIVO a la inyección de Caldo)
Pero si por el contrario, el antecedente se incumple
Entonces esto contribuye con ciertas razones positivas (34%)
al peso global sobre la siguiente condición:
    ({12} Se pronostica el Efecto POSITIVO a la inyección de Caldo)

Translation of the example is as follows:

Generalized production rule "(R1) 10 → 12 (-20 34)":
If the following fact is established with complete certainty weight:
    ({10} original production is HIGH)
Then this contributes with some negative weight (-20%)
to the knowledge about the following fact:
    ({12} POSITIVE effect of the Soup injection is expected),
but if, on the contrary, the antecedent is false
Then this contributes with certain positive degree (34%)
to the global weight of the following condition:
    ({12} POSITIVE effect of the Soup injection is expected).

3 Conclusions

We presented a method for generation of explanations that allows for generation of natural language explanations of production rules in expert systems, does not depend on knowledge base domain, and is easy to implement. The method can generate varying explanations of the same rule because text fragments are randomly chosen from a group of synonymous expressions associated with a rule.

The system consists of explanatory rules written in procedural programming language, fragments of natural language texts that are inserted in text slots in the explanatory rules, and textual descriptions of facts taken from the knowledge base.

Use of explanation rules makes generation of explanations independent of the program code: to add a new rule one does not need to add special code for generation of its explanation for the user.

References

[1] Alonso, M. A. y de la Cruz, A. V. Desarrollo de Sistemas Educativos Utilizando Técnicas de Computación Inteligente*. 1er Coloquio de Investigación en Ciencias de la Computación*. Memoria Técnica, pp. 73–79. 2001.

[2] Christian Boitet. Message Automata for Messages with Variants, and Methods for their Translation. In: A. Gelbukh (Ed.), *Proc. of CICLing-2005*, Spinger, 2005, pp. 349–368.

[3] Clancey, W. Neomycin: Reconfiguring a Rule-Based Expert System for Application to Teaching. In: *Proceedings of 7th IJCAI*, Vancouver, B.C., Canada, Morgan-Kaufmann, San Mateo, Calif., 1981.

[4] De la Cruz, A. V., Valdés, J. J., Jocik, E., Balsa, J. y Rodríguez, A. *Fundamentos y Práctica de la Construcción de Sistemas Expertos*. Editorial "Academia", Cuba, 1993.

[5] Giarratano, J., and Riley, G.. Expert Systems: Principles and Programming. Boston: PWS, 1998.

[6] Luger, G. F. and Stubblefield, W. A. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Third Edition, Addison-Wesley, 1998.

[7] Wick, M. R. and Thompson. Reconstructive Explanation: Explanation as Complex Problem Solving. In: *Proceedings of 11th IJCAI*, Morgan-Kaufmann, 1989.

[8] Winograd, T. A. Computer Program for Understanding Natural Language. *Technical Report TR-17*, MIT Artificial Intelligence Laboratory, Cambridge, Mass., 1971.