E-learning from Expertise: a Computational Approach to a non-textual Culture of Learning

Giovannina Albano, Ferrante Formato
DIIMA University of Salerno
Albano.Formato@diima.unisa.it

Abstract

The importance of “culture of learning” has been pointed out by many authors and is an emerging feature of e-learning. Our basic assumption, as Bruner has pointed out, is that learning is fully embedded in the cultural roots of the community in which people live. In this paper we propose a computational model of learning from expertise based on the theory of approximated reasoning developed by Gerla in [2] and on the studies on language made by Vygotskij and its school ([6], [5]). In our interpretation, expertise is an ability acquired mostly by experience. Our research is also supported by a software prototype that we are developing at University of Salerno, that we use for methodological purposes.

1. Introduction

Learning is mostly a socio-cognitive activity and therefore it is deeply connected with the culture of a community. Culture and cultural context deeply influence the way learning takes place, both through the way people communicate and represent knowledge, through the contents that are taught/learnt. This manner of learning is the most suitable for learning complex and highly structured notions, such as bridge or concrete structure design, control systems and even theoretical subjects such as mathematics and physics itself. None on the less, by identifying learning with “institutional learning” one could mistakenly identify learning with “just” one type of culture of learning: i.e. the “textual culture”.

It is a matter of fact that in early times, and still at presents, a craftsman gained his skill not much by studying on a book but rather by improving its abilities through experience. As an example, the Gothic Cathedrals in North-West part of France were built up by very skilled carpenters that where not used to codify their knowledge into books.

Still, today we often are confronted with tasks that can be learnt only by a simple system of rules and a lot of “expertise”, such as driving a car or playing soccer.

In this paper we formulate the hypothesis that this kind of learning is deeply relying on some properties of natural language that, at present, have not been fully investigated. As stated by Vygotskij’s school, the interplay between a word and its meaning its crucial, even though, in our case, the nature of this connection appears to be thoroughly peculiar. We observed that, unlike its institutional counterpart, learning by expertise is an adaptive process, and therefore it needs to be supported by a flexible language, where the meaning of words can be “tuned up” according to the particular context in which they are used. As a matter of fact, this features are embodied by natural language, that currently deal with vague notions.

The theory of approximated reasoning ([2], [7]) seems to be a suitable tools for an analytical investigation of this subject. As it is common in cognitive science, we have also formulated a computational model and we have developed a system called SciLog, an Integrated Development Environment that has been built at University of Salerno using Mathematica and Visual Basic. We have chosen these two systems because they combined rapid prototyping with a rich and handy mathematical library.

2. A computational model for learning by expertise

Approximated reasoning has been developed in the framework of fuzzy logic for control theory ([7], [4]) and has been recently given an adequate inferential apparatus by Gerla ([2]).

We send the reader to [5] for the terminology of logic programming. Given a set $X$, a fuzzy set, or a fuzzy subset of $X$, is a map $s:X\rightarrow[0,1]$. 
We now consider an example of fuzzy program. Let X=[0,30], Y=[100,600] and let the following fuzzy sets be given:

\[
\begin{align*}
\text{Cold}: X &\rightarrow [0,1], \text{Mild}: X &\rightarrow [0,1], \text{Hot}: X &\rightarrow [0,1] \\
\text{Slow}: Y &\rightarrow [0,1], \text{Moderate}: Y &\rightarrow [0,1], \text{Fast}: Y &\rightarrow [0,1]
\end{align*}
\]

We now introduce a new predicate name, as an example the predicate "Good" and we interpret \( \text{Good}(x,y) \) as:

"if the input variable is \( x \), then \( y \) is good value, according to the expert, for the control variable".

We now consider the following fuzzy program

\[
\begin{align*}
\text{Cold}(x) \land \text{Slow}(y) &\rightarrow \text{Good}(x,y). \\
\text{Mild}(x) \land \text{Moderate}(y) &\rightarrow \text{Good}(x,y). \\
\text{Hot}(x) \land \text{Fast}(y) &\rightarrow \text{Stable}(x,y).
\end{align*}
\]

This program represents a control of a cooling system of a PC; the set \( X \) represents the values of temperature of the CPU and the set \( Y \) is the domain of velocity of cooling fan. Let \( x \in X, y \in Y, \lambda_1 := \text{Cold}(x), \lambda_2 := \text{Slow}(y) \). Let \( \otimes \) be a t-norm, i.e. a binary operation on \([0,1]\) that is commutative, associative, monotonous and such that \( x \otimes 1 = x \). For example, in this case, we choose the t-norm of the minimum. We now describe an example of approximated reasoning:

\[
\begin{align*}
\text{Cold}(\hat{x}) \land \text{Slow}(\hat{y}) &\rightarrow \text{Good}(\hat{x},\hat{y}) \land \lambda_1, \lambda_2 \\
\text{Cold}(\hat{x}) \land \text{Slow}(\hat{y}) \land \text{Cold}(\hat{x}) \land \text{Slow}(\hat{y}) &\rightarrow \text{Good}(\hat{x},\hat{y}) \land \lambda_1 \land \lambda_2 \\
\text{Good}(\hat{x},\hat{y}) &\rightarrow \text{Stable}(\hat{x},\hat{y}) \land \lambda_1 \land \lambda_2 \land 1
\end{align*}
\]

I.e., in the case \( \otimes \) is the t-norm of the minimum, if \( \hat{x} \) is cold at least with a degree \( \lambda_1 \), and \( \hat{y} \) is slow at least with degree \( \lambda_2 \), then the two values are good at least with degree \( \text{Min}[\lambda_1, \lambda_2] \).

By repeating the reasoning for the other three rules, we can derive the predicate \( \text{Good}(x,y) \). In fact, for any \( x,y \) in \( X \) and \( Y \), respectively, we have:

\[
\text{Good}(x,y) = \text{Max}[\text{Min}[\text{Cold}(x), \text{Slow}(y)], \\
[\text{Min}[\text{Mild}(x), \text{Moderate}(y)], \\
[\text{Min}[\text{Hot}(x), \text{Fast}(y)]]).
\]

Once the predicate \( \text{Good} \) is known, and we have seen that it is possible to do so in a computational manner, it is possible to derive the ideal function \( f: X \rightarrow Y \) that is supposed to regularized to cooling system of the PC and can be computed, when it is possible, with the methods developed by classical calculus.

3. A case study: Learning to control a System by Expertise

In this section we describe the use of Scilog to model how an agent, either human or artificial, succeeds in mastering a control system, learning how it works by the use of his expertise.

Consider the problem of the regulation of temperature in a PC described in the previous section. We assume that, by a series of experiments and measurements, the expertise acquired by the agent about domains \( X \) and \( Y \) is represented by the families of fuzzy set of Figure 1. Suppose that, by another set of experiments, the agent learns that the “laws” that manages the control system are represented by the set of rules (*). In our case, the predicate \( \text{Good} \) stand for “stability” of the system according to the experience of the agent.

Then Scilog infers, i.e. computes the fuzzy predicate \( \text{Good} \) and this represents the expertise that the agent has learnt about the working of the control system.

We remark that in this process of learning knowledge has not been transmitted as something that is existing in an objective manner, but has been “built” in a constructive manner by the learner. We argue that, as stated by situationist theories, in a typical learning process learning is, de-constructed negotiated and re-constructed between a learner and a teacher, but this will be the subject of another work.

Bibliography