The Baby Webmind Project

Ben Goertzel, Ken Silverman, Cate Hartley, Stephan Bugaj, Mike Ross
Intelligenesis Corp.; 50 Broadway, NY NY 10004
ben@intelligenesis.net, ken@intelligenesis.net, cate@intelligenesis.net,
stephan@intelligenesis.net, miro@intelligenesis.net

Abstract

The Baby Webmind project involves teaching an instance of the Webmind AI system how to perceive, act and cognize through interaction in a shared perceptual environment. This talk describes the goals and methodology of the project at a high level, and briefly reviews some of the AI technologies underlying it.

1 Introduction

Webmind is an original AI architecture based largely on the interdisciplinary model of the mind developed in Goertzel (1994, 1997). It is based on a vision of the mind as an evolutionary, self-organizing, self-producing system. It is integrative in nature, involving reasoning, perception, cognition, evolution, long and short term memory and other modules embedded in a single dynamic data structure which allows emergent learning amongst the different components.

The goal of the project is not to simulate a human mind or to create a program that can pass the Turing test, but rather to create a software system with a mind. Our working definition of a mind is: a collection of patterns that forms and perceives patterns in itself and the world, in order to achieve complex goals in a complex environment. In Webmind's case the complex environment is the Internet.

The process of engineering Webmind has been ongoing since late 1997, and has now, thanks to the efforts of more than 40 AI engineers, reached a point where the major components of the system are complete and the focus can be placed on the integration of components and the induction of emergent intelligence in the whole system. This involves, among other things, a process of educating Webmind by interactive learning in a shared environment what we call "bringing up Baby Webmind."

2 Webmind

Webmind is a massively parallel network of static and dynamic information agents that continually recompute their relationships to other agents, implemented as a distributed Java software system.

The information agents populating the massively parallel self-organizing network that is Webmind come in many different species, each one specialized for different purposes. And all the different kinds of agents can learn from each other – the real intelligence of Webmind lies in the dynamic knowledge that emerges from the interactions of different species of agents.

There are numerical data processing agents, that recognize patterns in tables of numbers, using a variety of standard and innovative algorithms.

There are text processing agents, that recognize key features and concepts in text, drawing relationships between texts and other texts, between texts and people, between texts and numerical data sets.

There are reading agents, which study important texts in detail, proceeding through each text slowly, building a mental model of the relationships in the text just like a human reader does.

There are textual-numerical correlation agents, that recognize patterns joining texts and numerical data files together. These underly Webmind Market Predictor's unprecedented ability to find the concepts in news that drive the financial markets.

There are categorization agents of various kinds, that study the other agents in the mind, group them together according to measures of association, and form new agents representing these groupings.

There are learning agents, that recognize subtle patterns among other agents, and embody these as new agents. Among these are agents carrying out logical inference, according to a form of probabilistic logic based on Pei Wang's Non-Axiomatic Reasoning System (Wang, 19XX); and agents carrying out evolutionary learning, according to genetic programming (Koza, 1992), a simulation of the way species reproduce and evolve.

There are agents that model users' minds, observing what users do, and recording and learning from this information. There are agents that moderate specific interactions with users, such as conversations, or interactions on a graphical user interface. And there are self agents, that help Webmind study its own structure and dynamics, and set and pursue its own goals.

Each of these agents, in itself, has a small amount of intelligence, similar in some cases to that of competing AI products. The Webmind architecture provides a platform in which they can all work together, learning from each other and rebuilding each other, creating an intelligence in the whole that is vastly greater than the sum of the intelligences of the parts.

3 Teaching Baby Webmind

Even with all these diverse capabilities, the Webmind we have today is only a baby, exploiting 10architecture. To get all the mind modules to work together really intelligently, we need to lead the system step by step through goals, beginning with simple goals and gradually moving to more complex ones. We need to teach the system step by step almost like a human child.

Each of Webmind's modules is best of breed in some particular area, and the modules can be used independently or in various combinations to support various product functions. Putting a few modules together can give you functions that normal AI software can't do things like using text to predict the markets, or effectively filtering news messages for relevance. But putting all the modules together can get you actual intelligence, because the modules are chosen specifically so as to allow the system to understand itself, to recognize patterns in itself. To teach our baby Webmind, we plan to chat with it on a simple graphical/textual user interface. We won't chat with it about trees and flowers and teeth, because it doesn't have direct experience of these things. We'll chat with it about data files and shapes and MIDI music files, because these are the things that we can both experience. Intelligence has to be gained through interactive experience in a shared environment.

It's intriguing to see how the basic task of learning to interact in the world uses all Webmind's specialized modules. Reasoning and genetic programming evolution are used to find schema – sets of basic procedures for seeing and doing and thinking that are useful at achieving the system's goals and hence make the system happy. Categorization is needed to define contexts in the world a schema has to be judged by how it achieves important goals in relevant contexts. Language processing is obviously needed to chat with humans, and although in this context most of the specific nature of human language must be learned, nevertheless the basic structures needed for language understanding need to be provided from the start. Data processing is needed to turn raw numerical data files, sensed by the system, into comprehensible perceptual features. And so on. All the pattern finding and relationship building methods of Webmind's various modules are needed to provide the data that basic behavior schema need to act intelligently.

Acknowledgements

The ideas underlying this work owe a great deal to other members of the Intelligenesis AI staff, including especially Pei Wang, Jeff Pressing, Karin Verspoor, Anton Kolonin and Cassio Pennachin.

References

Ben Goertzel. *Chaotic Logic*. Plenum Press, New York, 1994.

Ben Goertzel. *From Complexity to Creativity*. Plenum Press, New York, 1997.

John Koza. *Genetic Programming*. MIT Press, Cambridge, MA, 1992.

Pei Wang. *Non-Axiomatic Reasoning System*. PhD thesis, Dept of CS & Cog. Sci. Indiana University, 1995.