Artificial Intelligence: Programming and Creation of Logic

Kevin McVey

The Commonwealth Governor's School

June 2, 2008

Abstract

(RESEARCH QUESTION) The purpose of this culminating project is to determine and discover the boundaries of the programing of computer logic and artificial intelligence. (FINDINGS) There are many different kinds of logic processes that are combined together under the term "artificial intelligence." These different logic processes range from trial and error style logic, to simply picking random numbers. The programming for any more than this, and the move into logic based upon what a computer "learns" becomes much more difficult. (PRODUCT) Because of the fact that the field of highly developed knowledge of artificial intelligence is somewhat small, this study tests the effectiveness of some of today's strategies for computer logic. To do this, this study created multiple forms of computerized logic that are all programmed into an artificially intelligent Tic-Tac-Toe game that upon playing 10 games with a user becomes "unbeatable." To show how different thought processes are tested, this product contains a special section on the user interface that allows the user to watch what the computer thinks about when it decides upon a move. (RESULTS) The product showed that reasoning based upon cases was quite difficult and complex to program, however, the most effective in aiding to the intelligence of the program. Aside from this, all data that was not clearly defined for the computer ("fuzzy" information it is called) became the most difficult to program and required much more thought. The type of reasoning that requires unclear criteria and information is around where our boundaries today are. (CONCLUSION) More powerful technology with the ability to comprehend data that is highly unclear and requires a greater level of reasoning is necessary before another large breakthrough in the area of artificial intelligence is to be made.

<u>Literature Review</u>

Introduction

As far back as the 1800's, science fiction has portrayed sophisticated machines capable of seemingly "human" thought processes, allowing non-humans to perform human tasks ranging from being slaves to showing affection. And ever since then, our imaginations combined with the current advances into technology, have shaped our dreams and fears of a sort of "thinking computer." Statistics from 2005 state that 98% of American homes have telephones, 50% have computers, and many families have multiple televisions, computers, cell phones, and other electronic devices. So, in our advancing state of technology, it seems more and more thought has gone into the development and creation of Artificial Intelligence (Thomas, 2005, p.6-18).

Artificial Intelligence is a collective term for many different sorts of logic, and the programming behind them. The literature on this subject tends to be specific and informative, and therefore requires many sources on each topic, as some tell "what" and others tell "how." Generally, authors write about the main divisions of programmed logic: Neural Networking/Neurology, Logic, and Programming/LISP. All of these break up into smaller sections, that different texts will dive into specifically, as each have equal necessity in the grand scheme of the technology.

Neural Networks & Senses

Neural Networks are a branch of Artificial Intelligence that one may call the beginning stage. These networks are better known as "computational models," or "parallel distributed processing." Neural Networks are more concerned with the hardware, and the ideals behind the intelligence, instead of being the actual program. In the human brain, the neural network is our mass of brain cells that constantly fire electric signals to other cells. In both the mind, and computers, the Neural Network is completely causes and effect, with the firing of electrons. And for those looking to recreate a sort of intelligence, the Neural Network is not to be passed over especially in the beginning stages, so that one may lay out thoughts in a graphic and organized manner. Years of research has gone into this, and the book by Ben Krose and Patrick van der Smagt show how specific and how broad this topic can be (1996). On a side note, however, much of this work may already be done, that is, if a programmer is using a modern PC and a standard/commonly used programming language.

Without a previously developed framework, a complex artificial intelligence will require senses. Our five senses we take for granted, because they are instinctive, however, a computer requires sensors and a large mathematical equation to produce output data that is useful. In today's world with growing demand for robots that perform special tasks, sensors are more important than ever. A walking robot would use rotation sensors to pinpoint its location and balance itself, and a driving robot would require a vision sensor so that it would know when to steer away from an impending obstacle. All of these different examples use matrices in a basic (x,y,z) Cartesian plot to generate useful data for the program to put to movement and practical use (Shah, 1997, p.5-20). While the average user may not expect to use anything near as complex as what is used in senses, it is important to understand how they work, to get a taste of the mind's internal logic processes.

Logic

The “logic and learning” aspect of the intelligence system can be broken up into five main parts: Machine Learning/Pattern Recognition, Fuzzy Systems, Evolutionary Computation, Case Based Reasoning, and Swarm Intelligence. Each has their own significance and when mixed together, they build the framework for a computer's ability to learn. It is this section that is on the front line of the program, interacting with the user, and comprehending what happens around it.

Machine Learning

Machine Learning and Pattern Recognition are the more “human” way of learning things. Quite simply, it's a large phrase for “trial and error.” The objective of Machine Learning is to learn by doing, such as when a child touches a hot stove and burn their finger; the child quickly learns not to touch it again. Machine Learning is just this, the user inputs something, and the system learns to respond in a specific way as a response to the input (Paliouras, Karkaletsis and Psyropoulos, 1998, p.9-20). This pattern, once learned can become the root of many future programs and thoughts by the computer as it tries to relate its current situations to previous patterns (McCarthy, 2007). This form of learning is the most basic, and fairly easy to program if given enough variables, however, it is just as important and useful as all others in that it is the most well known (and most often used) way of creating an intelligence database.

Fuzzy Systems

Fuzzy systems are a concept that are rather odd, and a bit difficult to grasp because the human mind works them out constantly without thought. Fuzzy logic is defined as the logic of uncertainty and vagueness, thinking where a regular linear computer could not. Today's computers require well documented and defined terms and sets; for example, if a user wanted a computer to scan a room and find a door, then before he or she was to start, this person would have to tell it exactly what a door looks like in excruciating detail. Otherwise, it would look over the door and search forever, or instead, it would get a false positive from a wall or something else in the room. Fuzzy logic is there to fill in the gray areas. Its purpose is to define the vague areas, when a computer can't use exactly perfect data. The literature on this topic ranges from the specifics above to the mathematical equations that run those processes (Dubois, 1980). Think of it like this; the term "beauty" is taken differently by pretty much everyone. A computer, would have no way of understanding this, because there is no absolute definition of beauty that everyone agrees on. So, the purpose of fuzzy logic is to make assumptions based on prior knowledge about other things on what is "beautiful." This can also be used for explaining the differences between different levels of something. For example, fuzzy logic would learn the uncertainty that separates a "high" grade from a "low" grade (Leake, 2002, para. 14-35). Fuzzy logic can be a bit shaky and buggy for programmers, but it saves a lot of time when the programming of an artificial intelligence becomes tedious.

Evolutionary Computation

While fuzzy logic is the process of uncertainty, evolutionary computation is the "natural selection" thought process that is a seemingly modified form of machine learning. Suppose a robot was in a grocery store that was on fire. It touches the fire, and learns that it is hot. However, this doesn't tell the robot never to touch anything hot again, it simply makes the urge to touch fire recessive, and the urge to run away more dominant. It is in this system, that the robot's dominating thoughts are recalled first for a specific stimulus, and if it does not achieve the expected outcome, it's mind moves down the list once more. So, if the robot find that running away doesn't help, and nor does climbing on top of the shelves, evolutionary computation says that it has tried every option, now it will have to jump through the fire to escape, despite the fact that it is a recessive and rarely recalled thought. It is completely based on theories, testing them, and coming back to previous data (Menon, 2004, p.1-25).

Case Based Reasoning

Even more complex and resource heavy is Case-Based-Reasoning. (AKA: CBR) CBR is a large system of case association, generally split into two sections: the case retriever, and the case reasoner. When a CBR program is run, it is given a problem case. This case is sent to the case retriever to find cases like it, while the case reasoner finds associations between the two, and multiple other cases. After combining all data, and finding all associations, a derived solution is found, and the CBR program ends. This is well outlined in the studies of Sankar Pal and Simon Shiu (2004). One could think of this like our judicial system. It looks up thousands of related topics, and combines them to find an answer. Think of it this way; if A1 = A2, A2 = A3, B1 = B2, B2 = B3, and A3 = B3, then by a sort of transitive property, A1 = B1. This is true because of the relation of all of the above topics. While it may not directly say that A1 = B1, it does say that A1 is the same as another topic, which equals another topic, and so on and so forth, until a final conclusion is met between the two strands of topics A and B. This, is Case-Based-Reasoning.

Swarm Intelligence

In the neurological studies of human thought by James Kennedy and Russel Eberhart, swarm intelligence is viewed as a sort of “brute force” attempt to solve a problem. Whether the inputs be from outputs from machine learning, or simple randomization, the purpose is to use every possible input to achieve the final correct output (2001). Swarm intelligence is commonly used by humans in our day to day lives, and with the correct randomization tools, fairly easy (however lengthy) to have a computer use as well. If a computer program is faced with the issue of finding “x plus seven equals 13” and was forced to use swarm logic, it would run through every possible value of x, until it comes to a correct answer. Swarm logic isn't efficient, but it is bound to find the correct answer eventually for nearly every situation.

Programming

A variety of languages are used in the development of artificial intelligence. The most popular was developed specifically for the production of artificial intelligence systems. This would be the programming language, LISP. While it is a bit of an older language, and a bit more specific than others, it is that language that is most advanced in its field. Other more generalized languages such as C++ and Java have been used in its place working fine, but not quite with the ease of a language created for the purpose of AI and logic.

LISP

The programming language, LISP, is a database type of language, meant for the ability to swap functions in and out, making it possible to create functions that happen only at specific times, or change the outcome in the middle of the program. The functionality of LISP is based upon conditionals and variable swapping (Chassel, 1990, p.1-23). The purpose of the system is to create an easy to use versatile programming language for dynamic programs. It is occasionally looked over in the creation of modern day programs, but is good to keep in ones mind when diving into the field of programming logic and databases (Touretzky, 1990, p.13-24).

Relevance

The present research was an outline of learning, and an attempt to grasp the massive amount of information that springs from the almost broad topic of Artificial Intelligence. The logic is generally based off the human mind, as if the brain is the final goal of AI developers. The creation and programming of a large intelligence for performing large important tasks requires lots of mapping out ideas, and lots of know how in the field of logic and mathematics. However, it is the consumer demand for such technology that keeps the business running and our understanding growing. It is understanding all of these different themes (programming, logic, mathematics, neurology/neural networks) that will in the end answer the previously stated research question, and pave the way to a final product.

Methodology

As an avid user of computers, I began with the notion that it would be a good idea to create a research project around th theme of computers and technology. Upon initial brainstorming of the project, I discussed the possibility of creating a sort of "artificial intelligence," seeing as that was a field that I really had not seen much information about. I really didn't have a large background in the field of artificial intelligence, but I had been programming small applications for my computer and building websites for a few years prior. This being said, I had a bit of programming experience before beginning the research process. From my ninth grade year to my tenth grade year, I had thought about my current knowledge of programming and did some research on artificial intelligence and the sort of logic that must be programmed to make an effective product. The original research question I created in the ninth grade was quite broad and difficult to answer. So, upon my research over my year since then, I changed my research question to better suit my needs, become less of a chore to answer, and become more of a product-driven question.

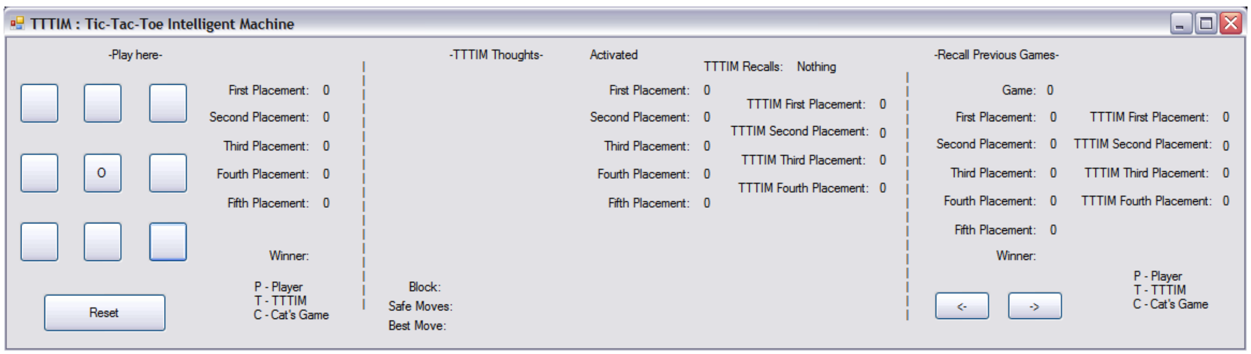
So, in a combination of some documents I gathered from a few public-access Internet libraries from Princeton University and Indiana University, I began outlining the project's literature review. The literature review helped me organize the information I had gathered. Unfortunately, this was far from enough information to work on a real product, so I went to my local libraries and picked up whatever books I could find on artificial intelligence, swarm logic, case-based reasoning, or other logic processes. After getting these books, I read through select chapters and began to understand what it really was that my project was going to require from me as a programmer. The entire process of gathering information turned out to be really more or less reading books and transferring important information into a literature review and the beginning of a product outline. All of the information I gathered was then measured against other books and the sources from the universities that I mentioned above to check for validity and a common standpoint between authors. Over the year as I learned more and more on the subject and became more and more interested, my focus shifted from business applications and the worldwide knowledge of artificial intelligence, to the actual “thought processes” of computers and the possibility of actual created logic.

My project was the culmination of multiple ideas of mine as I read through my sources. What I was truly aiming for was something that could accurately display what I had learned, but could also be fairly interesting for a user to use in a sort of “trial.” I had started with the thought that I could perhaps make a program to hold short conversation with the user, but eventually moved on to something a bit more interesting to my audience of students and school faculty. With this in mind, I came to the conclusion that a Tic-Tac-Toe game would fit my requirements. This game would be a program that could “learn” strategy and eventually become unbeatable after receiving various inputs through playing a set number of games with a user. Upon deciding this, I began looking at my options of programming languages. I decided that it would be best to use the .Net Visual Basic language for this project. I had been taking classes on this language all this year and have developed a strong understanding of this. To go along with this, I made a quick user interface that I believe may be a good base to begin working and testing off of. After narrowing my focus and working out an outline, I expected that this project would meet all of my requirements that I had set for myself and those that were set for me by my culminating advisers.

Upon completion of my project, I found that the goals I set for myself were very realistic and achievable. I managed to stick to my original user interface design with minimal changes. As the programming came to a close, I made a final comparison between my original goals and what I had completed. Upon many trials of the product, I was able to deduce that the product worked and my research was valid. Finally, I finished by adding a few finishing touches to the program's code and created a windows installer file to ease use.

Summary Of Results

This project was created to test the boundaries of artificial intelligence strategies and logic processes within today's technology. Upon creation of a strong graphical user interface with the ability to view the computer's thought processes, the product did just that. When given enough information, the product met its goal of becoming an unbeatable Tic-Tac-Toe intelligence and answered the research question as required.



The program's graphical user interface.

After all of the product's programming was complete and the program was tested, the pros and cons of each strategy became apparent. Different logic processes very quickly stood out while others faded into the back of the program's structure. Swarm logic was easy to program. It required very little thought, just plenty of room for error and plenty of time. Both of which, this program had neither. Therefore, though useful and highly productive, the use swarm logic was kept at a minimum.

The use of machine learning and evolutionary computation was effective. When both were put together, they were fairly easy to program and were very reliable. The only downfall to using them was the high amount of variables that they required, but this was easily recoverable with a few comments throughout the code. With very little error, the computer in this project was able to easily learn from different games that it played.

Case Based Reasoning (CBR) was the most time consuming to code, however, it proved to be a very reliable "safety net" so to speak when all other logic processes didn't quite work with a certain set of data. If all else failed, the program would hold current games up against ones it had already played and check for similarities. If any were found, the program would consistently pick the move it believed would be most beneficial to winning. However, this required the most tedious code in the product, and also required a large amount of variables. Neither of these were a large problem, simply minor setbacks in the grand scheme of time spent programming. Case based reasoning's pros in this case greatly outweighed the cons.

```
If g4ts3 = 6 Then
    lbld6.Text = "O"
End If
If g4ts4 = 6 Then
    lbld6.Text = "O"
End If
' ----
If g4s1 = 7 Then
    lbld7.Text = "X"
End If
If g4s2 = 7 Then
    lbld7.Text = "X"
End If
If g4s3 = 7 Then
    lbld7.Text = "X"
End If
```

Code from the CBR Section

Thought Process	**Difficulty Coding**
Fuzzy Logic	High. Very difficult and limited.
Machine Learning	Low. Easy and useful.
Evolutionary Computation	Low. Requires simple prior planning.
Case Based Reasoning	Medium. Repetitive and time consuming.
Swarm Intelligence	Low. Easy to program, easy to test.
LISP	Medium. Very little used, not too difficult.

The boundary that this project was aiming to find was found in the area of fuzzy logic. When the product was required to decide upon the certainty of something, it struggled. Only with the aid and implementation of LISP-like functions was the program able to overcome this setback. The code for the fuzzy logic section of the program became difficult to interpret and a bit sporadic. Due to the linear nature of computers and the nonlinear nature of the data used in fuzzy logic, the use of such logic was kept at very low levels.

The product overall was successful in achieving its goals. It discovered what logic strategies are and aren't practical on today's technology and discovered where the real boundary for our computers today lies. It interpreted all sorts of data, ranging from highly specific, to unclear. The results were all highly positive, and very clearly answered the original research question.

Suggestions for Further Research

While this product did bring up very clear results, there were a couple of areas that could have been further examined had more time been allotted. Although a pseudo-equivalent of LISP was created through the use of the "Public Function" operators within the code, LISP and Alisp were never never directly used. Not only this, but due to the difficultly of programming and low ability of computers to interpret unclear data, fuzzy systems were not used often. To further my research, it would probably be most beneficial to use more fuzzy systems, and an actual LISP machine.

WORKS CITED

Chassell, Robert. An Introduction to Programming in Emacs Lisp. 2nd. Boston, MA: Free Software Foundation, 1990.

Dubois, Didier, and Henri Prade. Fuzzy Sets and Systems. Chestnut Hill, MA: Academic Press, 1980.

Kennedy, James, and Russell Eberhart. Swarm Intelligence. San Diego, CA: Academic Press, 2001.

Krose, Ben, and Patrick Smagt. An Introduction to Neural Networks. 8th. The University of Amsterdam, 1996.

Leake, David. "Artificial Intelligence." Artificial Intelligence. 2002. Indiana University. 10 Jan 2008. <http://www.cs.indiana.edu/~leake/papers/p-01-07/p-01-07.html>.

McCarthy, John. "WHAT IS ARTIFICIAL INTELLIGENCE?." WHAT IS ARTIFICIAL INTELLIGENCE?. 2007. Stanford University. 10 Jan 2008 <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>.

Menon, Anil. Frontiers of Evolutionary Computation. New York, NY: Kluwer Academic Publishers, 2004.

Pal, Sankar, and Simon Shiu. Foundations of Soft Case-Based Reasoning. Haboken, NJ: John Wilkey & Sons, 2004.

Georgios, Paliouras, Vangelis Karkaletsis, and Constantine Spyropoulos. Machine Learning and Its Applications. Berlin: Springer-Verlag Berlin Heidelberg, 2001.

Shah, Mubarak. Fundamentals of Computer Vision. 1992. Orlando, FL: University of Central Florida, 1992.

Thomas, Peggy. The Lucent Library of Science and Technology: Artificial Intelligence. Farmington Hills, MI: Lucent Books, 2005.

Touretzky, David. Common Lisp: A Gentle Introduction to Symbolic Computation. Redwood, CA: Benjamin/Cummings Publishing Company, 1990.