High-level Design of SciBot: Architecture to Process Language and Execute Tasks

Morgan Ciliv

Aperture Laboratories
morganciliv@gmail.com

Introduction

The goal of the SciBot project at Aperture Laboratories is to create a robot that can be instructed through natural language to complete tasks such as microwaving Indian samosas and transporting objects such as warm Indian samosas, test tubes, chalk, and erasers to places or people.

SciBot's overall architecture is comprised of six separate, interacting modules, each implemented with unique classical and modern AI techniques. The architecture was constructed with simplicity and abstraction in mind. Inspiration is taken from human cognitive processes that occur during task completion.

The robot's six modules fall main components. The first is the input component which contains the language processing and visual object recognition modules which identify key components of their corresponding input. Second, the memory component, which is simply the memory module. The memory module incorporates the identified information from the input phase into the structure of the memory. Then, the planning component, which is the task planning module, takes those memories and the input to figure out what should be accomplished. Last, in the action component the agent uses all accumulated knowledge to figure out exactly what actions to take. The action component includes the path finding and motion planning modules.

The language processing module will parse the natural language sentences using probabilistic context-free grammars. The visual object recognition module will take in the array of pixels from the robot's camera and use a convolutional neural network to identify objects by name. The memory module will be an ontology and the task planning module will use a hierarchical task network to plan in the hierarchical world it will interact with. The path finding module will use A* search while the motion planning module which conducts movements for fine motor tasks will use a Bayes network. The modules can be visualized as shown below, where the arrows represent the flow of information.

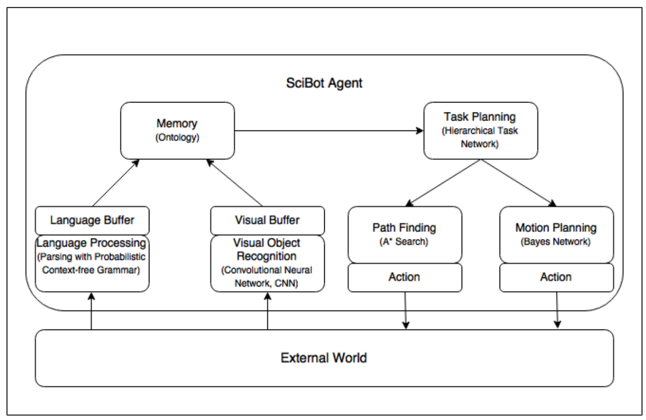

Figure 1: SciBot Architecture

Language Processing Module

The language processing module will take in sentences, a set of words, and try to decipher what the context of the word is in each sentence so that the words can then be used by the memory module for understanding.

Without the language module, it would be impossible for the robot to understand people's request. The robot needs to connect to what is going on in the external world to what it has in memory. This module helps solves incredible ambiguity that can exist, particularly in the English language.

The technique chosen for this module is a probabilistic (stochastic) context-free grammar (PCFG) which takes the concept of a context-free grammar but simply adds a probability to each rule. This technique is chosen because the incoming sentence needs to be parsed. If it weren't parsed then each of the words would simply be a listing instead of an interpretable instruction. More generally, parsing gives us structure so that the memory module can be indexed correctly. Furthermore, every natural language has a grammar and by programming in this grammar, the robot will be able to use the relationships of the words which is crucial to the sentence's meaning. Context-free grammars are a very straight forward way to parse a sentence because the structure and rule is directly encoded in them. A probabilistic free-grammar is an even better choice for this application because the natural language is full of uncertainties. The commander can easily make mistakes and vary time to time on how they express a command. A benefit of using PCFGs specifically to this architecture is that the model provides a definite structure which can be used by the ontology of the memory module, as described later.

There are drawbacks of using a PCFGs. As stated in Artificial Intelligence: A Modern Approach, "the problem with PCFGs is that they are context-free." Norvig and Russel then continue to explain how if similar sounding words could be fed into the model, the model wouldn't account for the context of the surrounding words. Another drawback mentioned by Norvig and Russel is that PCFGs work best for, not all, but particularly short sentences. However, this trade-off will be okay for SciBot as it mainly needs to understand simple commands.

The language processing problem will be modeled using this technique by defining the important elements of English grammar as a context free grammar. Before implementation, probabilities will be determined using statistical methods by testing in typical environments where SciBot will work. The model will then be oriented in the program such that it takes ouputs the structured data to the memory module.

Visual Object Recognition Module

This module will take in two-dimensional arrays of pixels from SciBot's cameras and determine from those two-dimensional arrays what objects are in its line of sight. This module is critical for SciBot's functionality because SciBot needs to be able to see what is around it to manipulate the external world. To safely and correctly act in the external world, an agent should have accurate input from the external world.

This module solves the problem of extracting information and drawing conclusions from data with great variation and uncertainty. This module steps into the realm of machine learning and computer vision techniques which completely step outside of simply programming a straight-forward, logic-based algorithm. These statistical methods are necessary to produce an accurate result. Our brains are the best object recognizers we know. Therefore, looking towards psychology and neuroscience can provide useful for inspiring a choice of algorithm. According to *How Does the Brain Solve Visual Object Recognition*, "evidence suggests that 'core object recognition,' the ability to rapidly recognize objects despite substantial appearance variation, is solved in the brain via a cascade of reflexive, largely feedforward computations that culminate in a powerful neuronal representation in the inferior temporal cortex."

Inspired from these feedforward neuronal processes in the cortex, convolutional neural networks [1](CNNs) will be used. We will use the model in SciBot particularly because of the method's success in object recognition problems. A convolutional neural network is a type of artificial neural network, so below I discuss the properties of neural networks more generally.

The benefits of using CNNs are that they have the highest accuracy rate. For example, in recognizing handwritten digits, CNNs have reached accuracy rates greater than approximately 99.7%. Handwritten images are a much easier problem to solve than real-world three dimensional objects; however, CNNs still scale. Another benefit of using CNNs is that they do not need to be explicitly programmed, the main additional work to be done is to train the neural network till we get a sufficient accuracy value on testing data sets.

The cons of using CNNs is that the model will have to be trained many times, and to teach SciBot how to recognize objects will have to do the training process all over again. However, there is no current computational model that does learn from small sets of data to the accuracy of CNNs.

The problem will be modeled using CNNs by having the two-dimensional image fed into the inputs of the neural network. The neural network will be composed of artificial neurons which fire once a threshold which fire only if their corresponding inputs are triggered to a high enough degree. After the training process, the neural network determines weights for certain characteristics of the image. Beyond this, the neural networks provide the abstraction that we don't need to know what's going on exactly.

[1] Google is trying to create a search algorithm based on convolutional neural networks that would allow us to take a picture of an object so that the computer could tell us what the object exactly is.

Memory Module

The memory module will take in the language processing's tagged words and sentences and the visual object recognition module's identified objects as inputs. It will then use that information to index the memory to identify what tasks need to be accomplished. This module will allow SciBot to make sense of the external world and to combine the information gained from two input streams so that SciBot can then plan and act on the world.

The memory module will be represented by an ontology. An ontology is a hierarchical model that represents is-a relationships between objects. By using the structure of the PCFG in the language processing module, the higher part of the ontology hierarchy can be the different parts of speech so that the words can be correctly identified and converted to logical predicates and variables which will be used in the planning module. However, the ontology will also further tell us where the object will fit into the context of the plan. For example, in the ontology there could be a parent node representing "things that I should deliver to," and under that class people and places will be stored as SciBot delivers objects to specific people or places.

The hierarchical structure of an ontology will bode well for an AI agent that is acting in the real-world because the world is naturally hierarchically structured. For example, SciBot will often be moving around in a building. In a building, there are floors and within those floors there are multiple rooms and within those rooms there are multiple people and objects such as tables. This hierarchically structured memory will allow the hierarchical task network in the planning module to be constructed more easily. Another benefit of this model is that objects can be placed occur under more than one class, which makes the representation more accurate.

A con to using an ontology is that the hierarchical structure and rules must be programmed in, and no one static model will model SciBot's environment perfectly. Furthermore, ontologies are not flexible and don't change based on experience or learning as human brains do.

Overall, as stated above, the ontology will be implemented through identifying characteristics and observations of the structure of the real-world. The model will be improved by trial and error usage of experiments done with SciBot.

Task Planning Module

The Task Planning module's goal is to formulate a plan and set of actions to take under certain conditions. This module solves the problem of deciding what SciBot should do next. This module lays out the instructions or physical tasks to complete so that the next modules can more specifically decide exactly what physical motions to take. The task module is critical to the role for a robot such as SciBot that needs to follow execute instructions. Because the actions couldn't be carried effectively without a plan. This module solves the problem of how to disseminate the instructions given from the input and produce the desired result.

Hierarchical Task Networks implement this planning module because of the hierarchical nature of the tasks which SciBot must accomplish such as making coffee and delivering the coffee to certain people on different floors in a building. Furthermore, SciBot and almost any robot will be repeating certain procedures, and the modular structure of an HTN allows for subtasks which can be called by various tasks.

A significant drawback to using HTNs is that the model doesn't incorporate uncertainty. However, SciBot will typically be active in buildings, which have low few structural changes occurring, and one of SciBot's main uses for this planning module will be used for planning where SciBot needs to be.

The HTN will be implemented in SciBot such that the predicates, variables, and values can have a direct correspondence with the classes in the ontology of the memory module such as actions, types of objects, and certain people or places. Many different possible actions predicates will be defined as SciBot will have various possible actions. An example of a task hierarchy for SciBot is the action go to a certain room. However, under the hood of that action will be "go to a certain floor of the building," and under that "go to the nearest elevator." Overall, the HTN will allow and be built for flexibility under a mostly deterministic environment.

Path Finding Module

The path finding module has the clear goal of find a path and particularly an efficient path from SciBot's current location to the desired location. SciBot will need this module because one of its main features is to deliver objects to people or places. Once the path is computed, SciBot will execute its movement based off the determined path.

The artificial intelligence technique chosen for this module is a classical and widely-used AI algorithm called A* Search. A* search was chosen because it is highly accurate and it has good average and worst-case time complexity. Another benefit to A* search over a more naïve search or path-finding algorithm is that it avoids obstacles more effectively.

A draw-back to using A* search is that better algorithms exist when one can pre-process the graph. However, SciBot doesn't have isn't able to preprocess a graph structure. SciBot will be mainly using the path finding algorithm for finding localized paths to take. SciBot will use it's be drawing location information on the fly. From a human's perspective A* search is still naïve as it doesn't look ahead at where it's going and plans accordingly. Furthermore, A* doesn't learn from experience, it will do the exact same path given the same data. That is where machine learning techniques can make improvements over A*. As another pro, A* doesn't need to be changed or altered, and is widely applicable to many searching domains which SciBot would need.

From the locations and the representation of SciBot's environment in memory, SciBot would determine an efficient path using A* on its graph representation. Once the path is determined, SciBot, a wheel based agent, will drive to the location following the A* determined path.

Motion Planning Module

In the motion module, SciBot will decipher what movements to take based off conditional probabilities. The motion module will allow SciBot to make Coffee or grab a test tube in its robotic hand for example. The module will be directly integrated with its motor actions. Every time this module decides something, it moves. This module solves the problem of moving in a scope of the world that is highly uncertain. For example, the coffee cup, which was placed by a human, could be in various places. Furthermore, the process of making coffee has great uncertainty because fluids and small coffee particles move quite stochastically.

Because of this high uncertainty in the scope of manipulating objects, the probabilistic graphical model of Bayes' networks has been chosen to implement this module. A Bayes' network uses given information and set conditional probabilities to identify what action should be taken next. The states of the network will represent the movements of the arm or hand that SciBot will take. As SciBot moves, the certain states will be set to a certain value so that the probabilities can be reevaluated.

The positives to using a Bayes network are not only that it incorporates uncertainty but also that it can update its model of those uncertainties based off conditions and values determined from the environment. For example, this would be useful for SciBot if SciBot clashed into something. The Bayes network would be defined such that if SciBot has hit something, the probability that SciBot should move in another direction is greatly increased over continuing in the wrong direction and making matters worse.

Bayes Networks must be programmed to take account of these uncertainties however. Since the world is uncertain, it's practically impossible to enumerate all variables and effects those variables should have on the model. Therefore, the actual Bayes' network implementation wouldn't account for many variables. Another aspect of implementing a Bayes network that should be noted is that the probabilities must be determined. Unfortunately, determining these probabilities requires training using machine learning techniques and field testing. SciBot's environment is constrained, so determining a reasonable number of conditional probabilities for the Bayes' network should be feasible.

Discussion

SciBot is designed to provide convenience for people. However, creating this intelligent agent is a very complex problem, and many actions that SciBot can take could easily be overlooked. Therefore, considering what ethical issues might come up with the implementation of SciBot in the workplace is critical.

First, SciBot poses a potential physical threat to the people. SciBot will be sharing its environment with people; therefore, physical harm could be an issue. SciBot will do exactly what its programmed to do, so there are ethical considerations regarding programmer is reliable about. Should we release SciBot if it doesn't account for people being in the way of it, or can we assume responsibility of the people? I think that SciBot should be built such that it moves slowly. As SciBot is demonstrated to be effective in the ethical space of physical harm, SciBot's speed can be increased if necessary. Making sure that SciBot moves slowly will also give time for the people to adapt to the new technology and for them to become aware of possible physical dangers. It's better to start incorporating these technologies into practice sooner than later so that we can all learn from the experience and better prepare for future, possibly more powerful technologies.

There are also many legal considerations regarding SciBot. As aforementioned, there is a duty upon the programmer to make the altruistic decision if given a choice between help or hurt. Moreover, we cannot enumerate all the possibilities. Therefore, are the programmer's legally liable for any destruction or ethical problem that the robot executes? There are also many other parties in question such as the company itself or the consumers that decide to use SciBot in the first place. I argue that each party is partially reliable. The blame cannot be put onto one party because each party made a role in the decision process. However, the system of the implementation of SciBot should be made so that there is extensive testing done before the implementation of the technology. Also, for the consumer, extensive training should be done with SciBot. This way producers and consumers are aware of the range of actions that SciBot can take will improvements to the entire, holistic process of implementation SciBot.

SciBot's implementation would cause a loss of jobs because the set of actions that SciBot can do play a role in how people make a living today. However, SciBot would simply be doing laborious work which only plays a minor role in jobs today. I think that SciBot's effect will be beneficial because it would raise the quality of jobs instead of laborious ones. Again, I think that the slower introduction process of SciBot would be better, in this case for economic considerations. I believe this because the people who have these jobs being replaced could have more time to make the transition. Also, these people could play a role in observing and making sure that the robots are taking the right action in terms of physical, privacy, and prejudicial issues.

When recognizing people, SciBot could accidentally be prejudice towards recognizing certain people because of their physical characteristics or data set. Therefore, testing should be done so that there is not an obvious bias in the data. Also, the dataset in which SciBot is trained on for its convolutional neural network should be a randomized sample and as large of a sample as possible without compromising the testing set data.

Overall, there are many ethical issues, in all ethical areas, that SciBot could potentially run into. However, if SciBot is implemented in a cautious manner where all people in the process are learning about the technology, then I think the number of ethical problems we would run into because of its implementation would be much less. Aperture Laboratories intention of benefiting humanity through SciBot would come to fruition.

Conclusion

In conclusion, the SciBot architecture will account for many variables in the building or area that it works. Scibot uses various modules to create a fully functional robot that has many characteristics like humans. SciBot takes in to account uncertainties but also uses logic to plan a course of action. SciBot is not self-conscious but it's course of action is not easily determined. There are many ethical problems that have not been enumerated. Implementation of SciBot should be proceeded with caution.

References

DiCarlo, J., Rust, N., and Zoccolan. Eds 2012. *How Does the Brain Solve Visual Object Recognition?* Neuron.

Norvig, P., and Russell, Stuart. eds. 1995. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ.: Pearson Education.