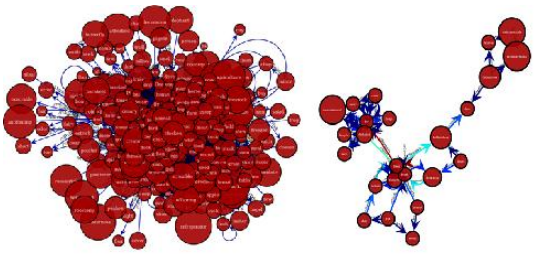
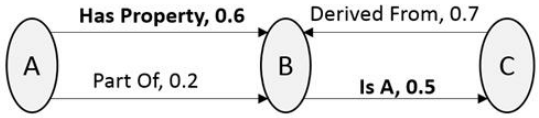
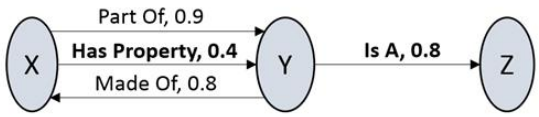
Contrasting Three Models of Analogy







eq = [A, "Has Property", B, "Is A", C]
ea = [X, "Has Property", Y, "Is A", Z]

Analogical reasoning is one of the most elegant forms of human reasoning. It requires making abstract connections, and is in fact quite difficult to get right all the time. What exactly makes an analogy work? Computer algorithms have been designed to model analogies in a variety of ways: some are used to distinguish what is the difference between a bridge and an arch, while others explain answers to difficult questions for humans to understand.

This project will first involve comparing, both qualitatively and quantitatively, a number of state-of-the-art and recent analogical reasoning algorithms (SME, LRA, and SSE). The second step will be to create an arbiter, which will decide which algorithm is best suited for a given problem, and try to establish a new performance record on existing benchmarks.

Some directions in which you could lead this project:

- Parallel text processing (one corpus of focus is 115GB in size)
- Graph search and subgraph identification (millions of nodes)
- Deploy these methods as web services for cloud robotics

Requirements: Python, NLTK, Machine Learning, Web services (optional)

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999
Credits: 4
Contact: Adrian Boteanu (ab2633@cornell.edu)

Arm Navigation for the KUKA youBot Mobile Manipulator

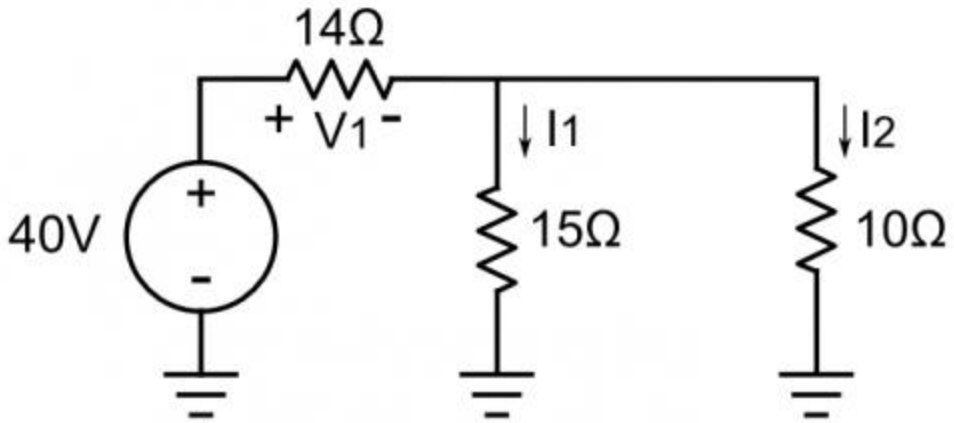
The KUKA youBot is an omni-directional mobile platform with an attached 5-DOF robotic manipulator. We are seeking highly motivated students to generate code that performs reliable arm navigation using the youBot's manipulator. The project involves the use of low-level navigation and manipulation functionality provided by ROS (Robot Operating System). We will perform extensive testing of the code and, by semester's end, expect it to be able to navigate to all points in its workspace with smooth, repeatable arm motion.

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999/ MAE4900/MAE6900
Credits: 3
Contact: Jon DeCastro (jad455@cornell.edu)

Graphical User Interface for circuit analysis: towards an auto-grader

The goal of this project is to create a graphical user interface that will allow students to record the process by which they analyze analog circuits (label currents/voltages, write equations, etc). This interface will be the basis of creating an auto-grader that will be able to provide personalized feedback for students learning about analog circuits.

Similar tool for learning how to construct automata can be found here:
http://www.automatatutor.com/



http://www.eeweb.com/electronics-quiz/basic-circuit-analysis-find-the-current-and-voltage

The project milestones are:

- Create a conceptual design for such an interface that would be appropriate for passive circuits, circuits with active elements and circuits with diodes. This design must capture all the steps of analyzing the circuit
- Implement the design so that it can run in a browser
- Set up a server to collect the information

Requirements: MAE 3780 or ECE 2100
Course number: MAE4900/MAE6900
Credits: 4
Contact: Prof. Hadas Kress-Gazit (hadaskg@cornell.edu), Prof. Erik Andersen (eland@cs.cornell.edu)

Collaborative behaviors for humanoid robots

In this project, you will work with two Aldebaran Naos on a multi-robot task. The goal of this project will be to develop coordination between two humanoids to carry an object from one place to another together, and possibly some other tasks that involve coordination. You will mainly focus on software development with the two humanoids.

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999/ MAE4900/MAE6900
Credits: 3-4

Contact: Catherine Wong (kw358@cornell.edu)

Assembly Task Design and Modeling for Robot Simulation

Robotics research can be greatly accelerated if algorithms that are tested in simulation can be then reliably applied in the physical world. In order to achieve this transition for manipulation tasks, such as pick-and-place or assembly tasks, the simulation environment must behave realistically and represent objects similar to those found in the real world.

The goal of this project is to design such simulation environments for testing and benchmarking robot algorithms focusing on assembly tasks. This requires

1. Designing tasks for real-world applications
2. Creating new object models.

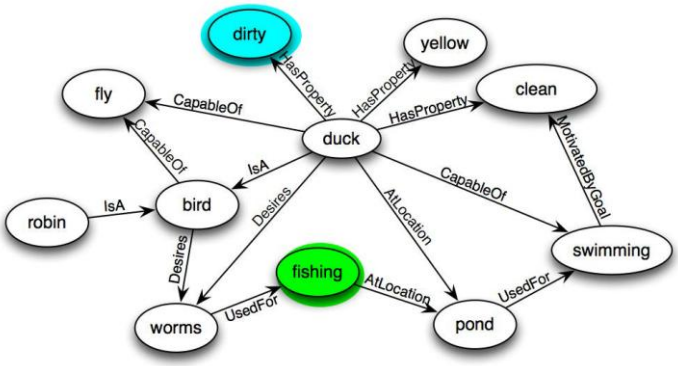
In addition to design skills, you will gain an understanding of real-world assembly operations as found in manufacturing, adapting them to the capabilities of the robot manipulator. The platform you will use is the Rethink Robotics Baxter. You will have access to a physical Baxter robot to deploy your designs. The simulation environment will be provided, it is custom code similar to Gazebo.

Requirements: ROS Unified Robot Description Format (URDF), 3D model design, C++

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999/ MAE4900/MAE6900
Credits: 2-3
Contact: Adrian Boteanu (ab2633@cornell.edu)

Comparison of Topic Modeling Methods for Recommendation Engines

Topic modelling algorithms associate words into cohesive groups. The words in each topic share a common theme, broad or narrow, and thus topics are particularly useful in applications such as search, document classification and generating suggestions. The prevalent approaches use Bayesian graphical models to infer topics from large document corpora. This results in good performance, but with results dependent on the training corpora. Alternatively, topics have been modeled using connectivity in semantic networks, shown in the picture.



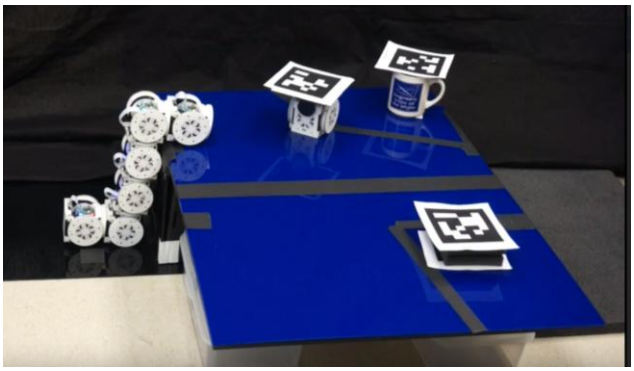
The goal of this project is to evaluate a variety of topic modeling algorithms over a number less common datasets, focusing on highlighting how algorithm performance changes depending on the application and corpus. Examples of dataset the project could use for evaluation are the Yelp Academic Dataset (https://www.yelp.com/academic_dataset), standard document collections, news transcriptions, and a dataset of dialog transcriptions between parents and their small children.

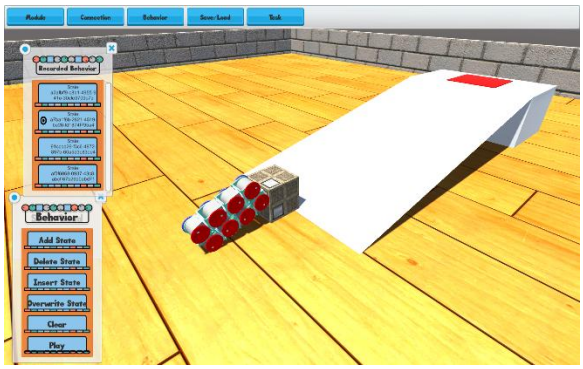
Requirements: Python, NLTK, Machine Learning, Web services (optional)

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999
Credits: 2-3
Contact: Adrian Boteanu (ab2633@cornell.edu)

Design and implement modular robot tasks in VSPARC

Do you like robots? Do you like to help make a computer game that programs robots? If you do, join us on the development of VSPARC. VSPARC stands for Verification, Simulation, Programming And Robot Construction. It is a game like toolbox made with Unity3d Engine. Users can design configurations and behaviors for our modular robot system. In addition, VSPARC also allows users to save and share their designs online. In order to encourage users to create robot behaviors with various abilities, we need to come up with different robot tasks frequently. Thus, we are looking for a student with experience with Unity3D Engine to help us design and implement robot tasks in VSPARC throughout the semester. In addition, the student is also encouraged to improve user experience of VSPARC by modifying the graphical interface, adding new features, and simplifying the design flow. For more information about the tool, please visit vsparc.org





Prerequisite: Experience with the Unity3D Engine

Professor: Hadas Kress-Gazit (hadaskg@cornell.edu)
Course number: CS4999/CS5999/ MAE4900/MAE6900
Credits: 3-4

Contact: Jim Jing (gj56@cornell.edu)