From Foundations to Current Work in a One Quarter Course on Artificial Intelligence

Michal Wollowski

Rose-Hulman Institute of Technology
5500 Wabash Ave.
Terre Haute, IN 47803, USA
wollowski@rose-hulman.edu

Abstract

We present and motivate the contents of a one quarter AI course. In this course, we provide a solid introduction to search and logic, combined with a philosophical review of classical AI. We furthermore introduce major tools and applications, including knowledge representation, reasoning under uncertainty and several forms of machine learning including neural networks. We end the course with student presentations of current work in AI.

Introduction

What should be the outcomes of a quarter long course on Artificial Intelligence? In this paper, we will present and motivate ours. We feel that students should develop a working knowledge of the fundamentals. In order to motivate the fundamentals, we feel that it is important that students get exposed to current work. Among others, this enables students to work on AI related senior theses, an important consideration in an undergraduate only institution. Covering the fundamentals and relevant current work does not leave much time for AI tools. We only cover them lightly, enough so that students are capable of understanding current work.

In order to solidify students' understanding of AI, we assign several review papers which question important aspects of AI. Students are conditioned to investigate AI applications so as to determine key features. We feel that this enables them to eventually work their way into AI applications which were not covered in the course.

In the remainder of the paper, we present some administrative aspects of the course, we present and discuss the major themes of the course and introduce the major assignments, grouped by programming assignments, reviews, and presentations of current work.

Administrative Aspects of the Course

It will be useful to have a quick look at some of the administrative aspects of this course. The course has been offered twice in its current form. It is offered during the fall quarter and we offer two sections with about 20 students enrolled in each of them. This represents 80% of a typical graduating class.

The Quarter System

We meet with our students four times a week for ten weeks. Each session lasts 50 minutes. Our student body is academically well prepared and as such, sessions are fairly intense.

Major Work

Students are assigned five major programming projects; currently they can work on one of them in pairs. Additionally, students review four key papers discussing philosophical issues of AI.

Programming Assignments	60%
Reviews	12%
Presentation of Current Work	15%
Homework Problems	8%
Participation	5%

Table 1: Weights of Components

Furthermore, students pair-up, study an active research project and give a class presentation about it. There are currently no exams and only minor homework assignments. The homework assignments are in support of the larger programming assignments. Due to the large amount of hands-on learning, there is no time for exams and it is not clear what should be tested on an exam. Table 1 indicates the weight of each grading rubric.

We typically have about three teaching assistants for the course. One teaching assistant usually takes the lead on developing the game assignment (see the section on

Programming Assignments). A significant part of their duties include the development of the support software which enables us to run the competition and to develop a simple competition player.

Resources

We use the Russell and Norvig [10] for background readings. In order to give non-logic based views of some of the materials and to fill gaps in coverage, we also assign papers, articles from the Wikipedia and other sources on the web.

Course Contents

We now provide details for the majority of the course contents. Consider table 2 which contains the schedule, see [20] for more details.

Overview

Based on our brief outline in the Introduction, there are three major themes in this course: fundamentals, applications, and current work. Integrated into this coverage are discussions of papers that raise important issues in AI.

We consider search and logic to be fundamental tools of AI. We spend about 12 days on them (days 1-10 and 15-16.) We split the coverage of logic as I was out of the country during days 11-14. We introduce logic as the study of valid reasoning, and motivate knowledge representation as ways to make valid reasoning more efficient. While the coverage of logic could be extended, it serves as a valuable motivator for the complexity underlying many AI applications.

We spend about 22 days covering major AI tools such as knowledge representation (days 11-13), planning (days 17 & 18), expert systems (days 19 & 20), game playing (days 21 & 22), and various machine learning techniques (days 23 – 34). While not all tools are relevant to current applications, as typically chosen by our students, their coverage represents major research activities of the past.

During days 32 and 37, fellow faculty gave guest lectures on their own work. This introduces our students to some of the research being done in our own department and is oftentimes used as advertisement of special topic courses in AI and to attract potential senior thesis students.

Day 1	Introduction to AI
Day 2	Searching
Day 3	Intelligent Agents, Turing Test
Day 4	Heuristic Search
Day 5	Properties of heuristic functions
Day 6	Making search efficient Introduction to Logic
Day 7	Proofs in propositional logic
Day 8	First-order logic
Day 9	Proofs in first-order logic
Day 10	Inference systems for FOL
Day 11	Knowledge representation Semantic networks
Day 12	Frames and the frame problem
Day 13	Scripts, Chinese Room
Day 14	Project Work Day
Day 15	Resolution, Unification
Day 16	Composition of Substitutions Resolution Strategies
Day 17	Introduction to Planning, STRIPS, partial-order planning
Day 18	Hierarchical planning, The great desert race
Day 19	Expert Systems
Day 20	Reasoning under uncertainty
Day 21	Introduction to Games
Day 22	Alpha-Beta Pruning
Day 23	Introduction to Machine Learning Inductive learning
Day 24	Knowledge in learning
Day 25	Project workday
Day 26	Learning by discovery Case-Based Reasoning
Day 27	Introduction to Case-Based Reasoning
Day 28	Case-Based Reasoning in more detail
Day 29	Data Mining
Day 30	Introduction to Neural Network Perceptrons
Day 31	Multi-layer Perceptrons
Day 32	Scene Recognition
Day 33	Other kinds of Neural Networks
Day 34	Introduction to Evolutionary Computing
Day 35	Finals of the game competition
Day 36	Student presentations of current work
Day 37	Evolutionary Computing
Day 38	Student presentations of current work
Day 39	Student presentations of current work
Day 40	Student presentations of current work

Table 2: Schedule

Motivation

We begin the course by presenting some of the early grand visions for AI. In this context, we assign our first programming assignment, a rudimentary version of ELIZA. This is a fun exercise and it lays a foundation to discuss the Turing Test and Searle's "Minds, Brains, and Programs" [11]. Students are asked to review both of those papers and we discuss them in class. I find that the reviews set a sobering perspective from which to evaluate what we cover in this course. While students work on ELIZA and the Turing paper [13], we review basic search and cover heuristic search. This leads to the next programming assignment in which students have to implement heuristic search. The problem that they need to solve is called

"Forest Fire Rescue." See the section on "Programming Assignments" for detailed information.

While students work on the heuristic search problem, we introduce the basics of propositional and predicate logic. We introduce logic as the study of valid reasoning and cover the basic rules of inference. Students practice the process of formalizing knowledge into a structured language as well as the process of giving formal proofs by working on several homework assignments. By the time we covered logic, students have completed their heuristic search assignment and we assign a resolution theorem prover. We now introduce some basic knowledge representation formalisms, including scripts. This enables us to discuss Searle's article, as mentioned above. We feel that Searle's paper and the homework assignment on formalization amplify each other well. By now, we reached a point that is far removed from the grand vision of AI and provides a sobering experience for our students.

It is now time to introduce students to actual AI applications. To get students excited about AI, we cover planning and ask our students to read excerpts of the paper about Stanley, the car that won the DARPA Grand Challenge [12]. We ask students to assess why this system is an AI system. While still sobering, Stanley is an achievement that students relate to well.

Soon after we introduce planning, we cover game playing. This sets us up for a component of this course that students always enjoy, the game playing competition. Students pair up to develop code for a computer game that ideally incorporates alpha-beta pruning and learning, although students have done well with brute-force solutions. We assign a different game every year and hold a competition during the last week of class in which the software systems compete against each other.

The next twelve days are spent on various forms of machine learning. We introduce the basics and provide background for some techniques for which colleagues give guest lectures. As part of our coverage of Machine Learning we also introduce neural networks. We introduce them as a potentially different way of accomplishing artificial intelligence and in this context assign a paper by Tim van Gelder [14] in which he suggests a way of explaining cognition that is not based on a formalization to symbol processing. By now, students have a very good sense of what is and is not possible in AI and this paper is a pleasant way to challenge their curiosity.

The remaining days are spent on one of the most pleasurable part of the course, student presentations of current work. Since students may choose the work they present, they typically spend a good amount of effort on this part and the results are quite pleasing. Asking students to present their work gives all students a broader perspective of current research.

In the remainder of this paper, we provide details on the primary assignments in this course.

Programming Assignments

The programming assignments are where the bulk of the learning in this course takes place. They are weighted heavily, counting 60% towards the course grade. There are five programming assignments; some require more work than others.

ELIZA

This is the first assignment and is intended to introduce students to an early goal of AI. It is due within three days and students may select any programming language. We show our students the ELIZA program built into emacs and point them to the pages of the Loebner Prize competition [8]. Students are asked to implement the random response and keyword response units, using at least 10 responses in each unit. We also ask for some very rudimentary parsing, in essence, recognizing first person sentences and turning them into second person responses. The assignment gives students a basis on which to discuss the Turing Test and Searle's Chinese Room argument (see the section on Reviews.) For more information, see [15].

Forest Fire Rescue

This assignment asks students to implement A* and to develop a rather complex heuristic function. The assignment is adapted from one given by Andrew W. Moore, in an AI course at Carnegie-Mellon University. Students are asked to develop a heuristic and a cost function for this problem and turn them in after a few days so that we can give our students feedback on them. A key aspect of this problem is that it requires in essence a two stage heuristic. We will present some details of the assignment here and refer the reader to [16] for more information.

In the Forest-Fire Rescue problem, the software control a set of trucks. The objective is to rescue people who are trapped in a forest fire in an expeditious manner. Trucks can move in an expected manner: up, down, left, or right by exactly one position per state transition. When a truck moves into a position occupied by a person, the person is automatically loaded on board and disappears from the map. Trucks may not move into a fire square, and an arbitrary number of trucks may move per turn. People cannot move and fires do nor grow or shrink. Distress is calculated by the inverse of the distance to the closest fire. When all people are picked up, trucks exit to safety by moving off to the left side of the grid.

Theorem Prover

Logic is in many ways the foundation of classical AI. We introduce logic as the study of valid reasoning. Knowledge representation and reasoning under uncertainty are introduced as making logic more efficient and as dealing with real world reasoning, respectively. As such, it is important that students have a good grasp of theorem provers, their limitations, and as such, limitations of AI software in general. We assign the implementation of a resolution theorem prover. Students have to implement a unification algorithm, resolution and a resolution heuristic of their choice. For more information, see [21].

Game Competition

Every year, we ask our students to implement some game. After the due deadline, we hold a competition which adds a good amount of motivation to our students. Students are allowed to work in pairs and this assignment is typically due near the end of the quarter, which in principle enables students to incorporate as much of the course materials as they like. This usually does not happen and some students did well by developing brute force solutions. We use such occurrences to discuss IBM's Deep Blue chess player.

This quarter, students were asked to implement the game of checkers [9]. Checkers is appealing because it is one of the earliest AI games and because it is easy to write a program that beats the programmer; demonstrating a good example of the power of AI. Additionally, we chose it as there is no temptation to be concerned with an elaborate GUI, something that distracted students in past game competitions. Typically, we run the programs against each other, using a game server. For further information, see [17].

Neural Networks

During the most recent offering of the course, we added an assignment on neural networks. For now, it is a rather rudimentary assignment. In the future, we plan to give students more time for this assignment and to make it more complex. Right now, we ask our students to set up a given neural network and train it. It is a two part assignment in which students' first familiarize themselves with the Joone software [4] for neural networks. In this part, they are asked to train a feed-forward network that implements the XOR boolean function. In the second part, students are asked to complete an assignment on character recognition [2] which is part of the Joone distribution.

In the future, we hope to ask students to design the network themselves. While working on this assignment, students also read a paper by van Gelder [14] in which he argues for a conceptualization of cognition that would best be implemented through neural networks. For more information, see [18].

Reviews

Students are asked to review four papers covering fundamental aspects of AI. We discuss all four papers after the reviews have been turned in. The papers enable our students to think about some of the goals and problems facing AI. We attempt to tie in the papers to the course contents and the assignments. Right now, there are no concrete guidelines for the reviews, except for the length. For the next offering, we plan to provide more guidelines, in particular, we plan to ask students to use the guidelines used by reviews.com. This gives students a chance to practice the art of the review and enables them to read the papers with a better focus. The reviews count 12% towards the final grade.

Turing Test

In parallel to working on the ELIZA programming assignment (see Programming Assignments) students review Turing's "Computing Machinery and Intelligence" [13]. We discuss the Turing Test and study transcripts of some of the Loebner Prize contestants. This combination works very well.

Chinese Room

After we had a look at knowledge representation formalisms, including Scripts, we read Searle's "Minds, Brains, and Programs" [11]. This paper prompts students to think about potential differences between human and machine cognition and puts KR in perspective.

Stanley

Around the time we cover planning, students are asked to review two sections of a paper about Stanley [12], the car that won the DARPA Grand Challenge. One section has to be the one entitled "Planning," the other section can be chosen at will. By this time in the course, we have abandoned the grand plan of building systems with human intelligence and we are focusing on real world systems. Stanley is a set of sophisticated AI applications in a context that is relevant and very accessible to students.

Dynamical Systems

When we cover Neural Networks, students are asked to read the paper "Revisiting the Dynamical Hypothesis" by Tim van Gelder [14]. van Gelder provides a view of cognition that is an alternative to the classical Turing Machine based conceptualization. Students find this paper very accessible.

Presentation of Current Work

This is probably the most important part of the course. At the very least it is the most enjoyable part for the instructor as they get to learn about interesting AI applications. While it is important for students to get a solid foundation in AI, it is equally important for students to see what sort of work is being done in the field and how people use the techniques introduced in the course. This serves as a motivating factor for students to enroll in advanced AI courses and to pursue senior theses in this area.

There are two components to this part, an oral presentation and a write-up. The presentations are scheduled for 20 minutes, about the same time as a conference presentation. This exposes students to the timing and rhythm of research presentations. We added the write-up because it helps students produce good quality presentations. This assignment has to be done in pairs, as this too significantly increases the quality of the work. The presentation and write-up are worth 15% of the final grade.

Students are asked to select a research project and are presented with listings of current work, such as: The AAAI List of Applications of AI [1], MIT [6], Stanford University [7], Carnegie Mellon University [5], and a site on game AI [3]. Students are encouraged to select projects from other sites as well.

We ask our students to summarize the work on which they report and to demonstrate existing software, if available and accessible. We encourage them to contact the principle investigator, suggesting that this may be an excellent way to determine potential graduate schools. In order to encourage students to reflect on what they have learned in the course, we ask that each presentation address the following questions: Does the project use searching? If so, what kind of searching and where? What formalism is used to represent knowledge? Why is this an AI project, instead of, say, an interesting data-structures project? What do you see as a pitfall in this project? How would you extend this work?

For more information, see [19].

Conclusions

We presented and motivated the major components of a one quarter course on Artificial Intelligence. Among others, we highlighted how we integrate class materials, programming assignments, reviews and the presentations of current work. In addition to introducing AI, we also use this course to implant in students' minds the possibility of attending graduate school, something our students traditionally do not consider. As such, some of the assignments and formats of assignments are designed to start preparing them for graduate level work.

References

[1] AAAI List of AI Applications, 2007. http://www.aaai.org/AITopics/html/applications.html.

[2] Character Recognition Neural Network Exercise, 2007. http://www.jooneworld.com/wiki/tiki-index.php?page=CharacterRecognitionSystem

[3] Game AI Pages, 2007. http://www.gameai.com/

[4] Homepage for Joone Neural Network Engine, 2007. http://www.jooneworld.com/

[5] List of Carnegie-Mellon Projects, 2007. http://www.ri.cmu.edu/project_lists/index.html

[6] List of MIT Research Activities, 2007. http://www.csail.mit.edu/research/activities/activities.php

[7] List of Stanford University Research Areas, 2007. http://cs.stanford.edu/research/

[8] Loebner Prize Competition, 2007. http://www.loebner.net/Prizef/loebner-prize.html

[9] Rules of Checkers, 2007. http://en.wikipedia.org/wiki/English_draughts

[10] Russell, S. and Norvig, P. 2003. *Artificial Intelligence, A Modern Approach, 2nd Ed.*, Upper Saddle River, NJ.: Prentice Hall.

[11] Searle, J.R. 1980. *Minds, Brains, and Programs*, The Behavioral and Brain Sciences 3, pp 417-424.

[12] Thrun S., et al. 2006. *Stanley: The Robot that Won the DARPA Grand Challenge*, Journal of Field Robotics 23(9), pp 661–692. Wiley Periodicals, Inc.

[13] Turing, A.M. 1950. *Computing Machinery and Intelligence*, Mind 59, pp 433-460.

[14] van Gelder, T. Undated. *Revisiting the Dynamical Hypothesis,* http://www.philosophy.unimelb.edu.au/tgelder/papers/Brazil.pdf

[15] Wollowski, M. 2007. *ELIZA Assignment*, http://www.rose-hulman.edu/class/csse/csse413/assignments/eliza.html.

[16] Wollowski, M. 2007. *Forest Fire Assignment*, http://www.rose-hulman.edu/class/csse/csse413/assignments/forestfire.html.

[17] Wollowski, M. 2007. *Game Competition*, http://www.rose-hulman.edu/class/csse/csse413/assignments/game.html.

[18] Wollowski, M. 2007. *Neural Networks Assignment*, http://www.rose-hulman.edu/class/csse/csse413/assignments/nn.html.

[19] Wollowski, M. 2007. *Presentation of Current Work Assignment*. http://www.rose-hulman.edu/class/csse/csse413/assignments/presentation.html.

[20] Wollowski, M. 2007. *Schedule*, http://www.rose-hulman.edu/class/csse/csse413/schedule/.

[21] Wollowski, M. 2007. *Theorem Prover Assignment*, http://www.rose-hulman.edu/class/csse/csse413/assignments/theoremprover.html.