Unifying an Introduction to Artificial Intelligence Course through Machine Learning Laboratory Experiences

Ingrid Russell, Zdravko Markov, Todd Neller, Michael Georgiopoulos,
Susan Coleman
University of Hartford/Central Connecticut State University/Gettysburg
College/University of Central Florida/University of Hartford

Abstract

This paper presents work on a collaborative project funded by the National Science Foundation that incorporates machine learning as a unifying theme to teach fundamental concepts typically covered in the introductory Artificial Intelligence courses. The project involves the development of an adaptable framework for the presentation of core AI topics. This is accomplished through the development, implementation, and testing of a suite of adaptable, hands-on laboratory projects that can be closely integrated into the AI course. Through the design and implementation of learning systems that enhance commonly-deployed applications, our model acknowledges that intelligent systems are best taught through their application to challenging problems. The goals of the project are to (1) enhance the student learning experience in the AI course, (2) increase student interest and motivation to learn AI by providing a framework for the presentation of the major AI topics that emphasizes the strong connection between AI and computer science and engineering, and (3) highlight the bridge that machine learning provides between AI technology and modern software engineering.

In this paper we will present our approach, an overview of the project, and the hands-on laboratory modules. Our preliminary experiences incorporating these modules into our introductory AI course will also be presented.

1. Introduction

An introductory Artificial Intelligence (AI) course provides students with basic knowledge of the theory and practice of AI as a discipline concerned with the methodology and technology for solving problems that are difficult to solve by other means. The importance of AI in the undergraduate computer science curriculum is illustrated by the Computing Curricula 2001 recommendation of ten core units in AI[2]. It is believed by many faculty members that an introductory AI course is challenging to teach because of the diverse and seemingly disconnected topics that are typically covered[6]. Recently, work has been done to address the diversity of topics covered in the course and to create a theme-based approach. Russell and Norvig present an agent-centered approach[21]. A number of faculty have been working to integrate Robotics into the AI course[3,7,8,9].

Our approach involves using machine learning as a unifying theme to teach fundamental AI concepts. Machine learning is used as a theme for two main reasons. First, learning is becoming an increasingly important area of computer science that is playing a major role in a wide range of applications. Second, the coverage of search algorithms in an AI course provides an ideal setting allowing us to easily expand such coverage to machine learning algorithms. Machine learning algorithms also provide excellent examples of heuristic approximation algorithms.

The National Science Foundation (NSF), through its Combined Research and Curriculum Development program, has recently funded a project involving the integration of machine learning into the engineering curriculum[4]. The project involved two phases, one that integrates machine learning modules into a variety of first and second year engineering courses and the second phase that involves the development of two upper level courses in machine learning. Our current project, also funded by the National Science Foundation, is an adaptation of the above project. Our target audience is different. Our material targets juniors and seniors who have a strong computer science background, including programming, data structures and algorithms, and discrete mathematics. Thus, we can concentrate on machine learning concepts and use them as a unifying theme for introducing the core concepts of artificial intelligence. In addition, the framework being proposed is adaptable to allow instructors to extend it based on local needs.

Our project incorporates machine learning as a unifying theme for the AI course through a set of hands-on lab projects. Machine learning is inherently connected with the AI core topics and provides methodology and technology to enhance real-world applications within many of these topics. Machine learning also provides a bridge between AI technology and modern software engineering. As Mitchell[12] points out, machine learning is now considered as a technology for both software development (especially suitable for difficult-to-program applications or for customizing software) and building intelligent software (i.e., a tool for AI programming).

Planning algorithms and machine learning techniques are important in several areas of AI and hence their in-depth coverage is important in such a course. While at times an agent may be able to react immediately, there are times where planning and evaluating potential actions is important. Learning, therefore, is a particularly important concern when building intelligent systems. In a similar way, computer systems and programs are limited by the designer’s or programmer’s limitations. Learning allows a system to adapt and improve its performance based on experience. Such applications are widespread in areas such as natural language processing, computer vision, and robotics, among others.

Our machine learning emphasis acknowledges that intelligent systems are best taught through their application to challenging problems. The deliverable will be a laboratory manual, with labs involving the design and implementation of a learning system which will enhance a particular commonly-deployed application. A broader impact of this project will be achieved through the collaborative development and separate testing of these labs at the three diverse participating institutions and through effective dissemination of this material to 21 other participating faculty members from academic institutions who have committed to using and testing these hands-on laboratory projects in their introductory AI courses.

To be widely adoptable, our work must be easily adaptable. While our approach will be experientially focused, we do not presume that a single project will have the right amount of implementation for every instructor's usage. Our designs will therefore be modular and object-oriented, allowing instructors to customize assignments. While we envision using such labs for a full semester of projects, the different entry points into each project will allow faculty to tailor these projects as they deem necessary for their course.

2. AI Course Overview

We have taught introductory AI courses several times. AI courses provide students with basic knowledge of the theory and practice of Artificial Intelligence as a discipline concerning intelligent agents capable of deciding what to do and doing it. Our offerings at all three institutions have been largely consistent with traditional offerings in providing students with basic knowledge of the theory and practice of AI as a discipline. The material covered includes search algorithms, knowledge representation and reasoning, as well as a brief introduction to several sub-fields of AI. The courses are taught at the junior/senior level and require Data Structures as a pre-requisite.

The course objectives are:

- To have an appreciation for and understanding of both the achievements of AI and the theory underlying those achievements.
- To have an appreciation for the engineering issues underlying the design of AI systems.
- To have an understanding of the basic issues of knowledge representation and blind and heuristic search, as well as an understanding of other topics such as minimax, resolution, etc. that play an important role in AI programs.
- To have a basic understanding of some of the more advanced topics of AI such as learning, natural language processing, agents and robotics, expert systems, and planning.

A sample course syllabus used based on Russell and Norvig book[21] is in Figure 1 below.

Introduction
- What is AI?
- Foundations of AI
- History of AI

Intelligent Agents
- Agents and Environments
- Structure of Agents

Problem Solving by Searching
- Problem Solving Agents
- Searching for Solutions
- Uninformed Search Strategies: Breadth-First Search, Depth-First Search, Depth-limited Search
- Iterative Deepening Depth-first Search
- Comparison of Uninformed Search Strategies

Informed Search and Exploration
- Informed (Heuristic) Search Strategies: Greedy Best-first Search
- A* Search

- Heuristic Functions
- Local Search Algorithms and Optimization Problems

Constraint Satisfaction Problems

- Backtracking Search for CSPs
- Local Search for CSPs

Adversarial Search

- Games
- Minimax Algorithm
- Alpha-Beta Pruning

Reasoning and Knowledge Representation

- Introduction to Reasoning and Knowledge Representation
- Propositional Logic
- First-order Logic
- Semantic Nets
- Other Knowledge Representation Schemes

Reasoning with Uncertainty & Probabilistic Reasoning

- Acting Under Uncertainty
- Bayes' Rule
- Representing Knowledge in an Uncertain Domain
- Bayesian Networks.

Learning

- Forms of Learning
- Decision Trees and the ID3 Algorithm
- Statistical Learning
- Summary of other Approaches

Figure 1: Sample Syllabus

The beginning of this course covers a brief introduction to the Lisp programming language. This is followed by problem-solving techniques including problem spaces, uninformed as well as informed search techniques, and the role of heuristics. Two-player games and constraint satisfaction problems are covered next along with planning techniques. The course then covers knowledge representation schemes including predicate logic, non-monotonic inference, probabilistic reasoning, production systems, semantic nets and frames. The last part, which in other courses typically consists of exposure to several AI fields, consists of approximately three weeks of coverage of machine learning concepts and algorithms. Students are also expected to write a paper on an AI topic not covered in the course and present it in class. This is their opportunity to research an AI area of interest and gain exposure to other AI fields. With the exception of the learning module, the course is based on Russell and Norvig[21].

3. Project Goals and Objectives

The difficulties mentioned above associated with the introductory AI course, combined with the increasingly important role of machine learning in computer science in general and software development in particular, are the motivating factors for our approach. The objectives of our project are listed below:

- Enhance the student learning experience in the AI course by implementing a unifying theme of machine learning to tie together the diverse topics in the AI course.
- Increase student interest and motivation to learn AI by providing a framework for the presentation of the major AI topics that emphasizes the strong connection between AI and computer science.
- Highlight the bridge that machine learning provides between AI technology and modern software engineering.
- Introduce students to an increasingly important research area, thus motivating them to pursue more advanced courses in machine learning and to pursue undergraduate research projects in this area.

These objectives are accomplished through the development, implementation, and testing of a suite of adaptable and self-contained, hands-on open laboratory projects that can be closely integrated into the AI course. Three diverse institutions are collaborating on this project. The University of Hartford is a mid-size comprehensive private institution, Central Connecticut State University is a large state institution, and Gettysburg College is a small liberal arts private institution. A broader impact of this project will be achieved through the collaborative development and separate testing of these labs at the three diverse participating institutions and through effective dissemination of this material to several other participating faculty members from academic institutions who have committed to using and testing these hands-on laboratory projects in their introductory AI courses.

The paper presents an overview of our project and some preliminary results of testing some of the material at the authors' departments. This paper reports on the first phase of the project which was accomplished during Summer and Fall 2004.

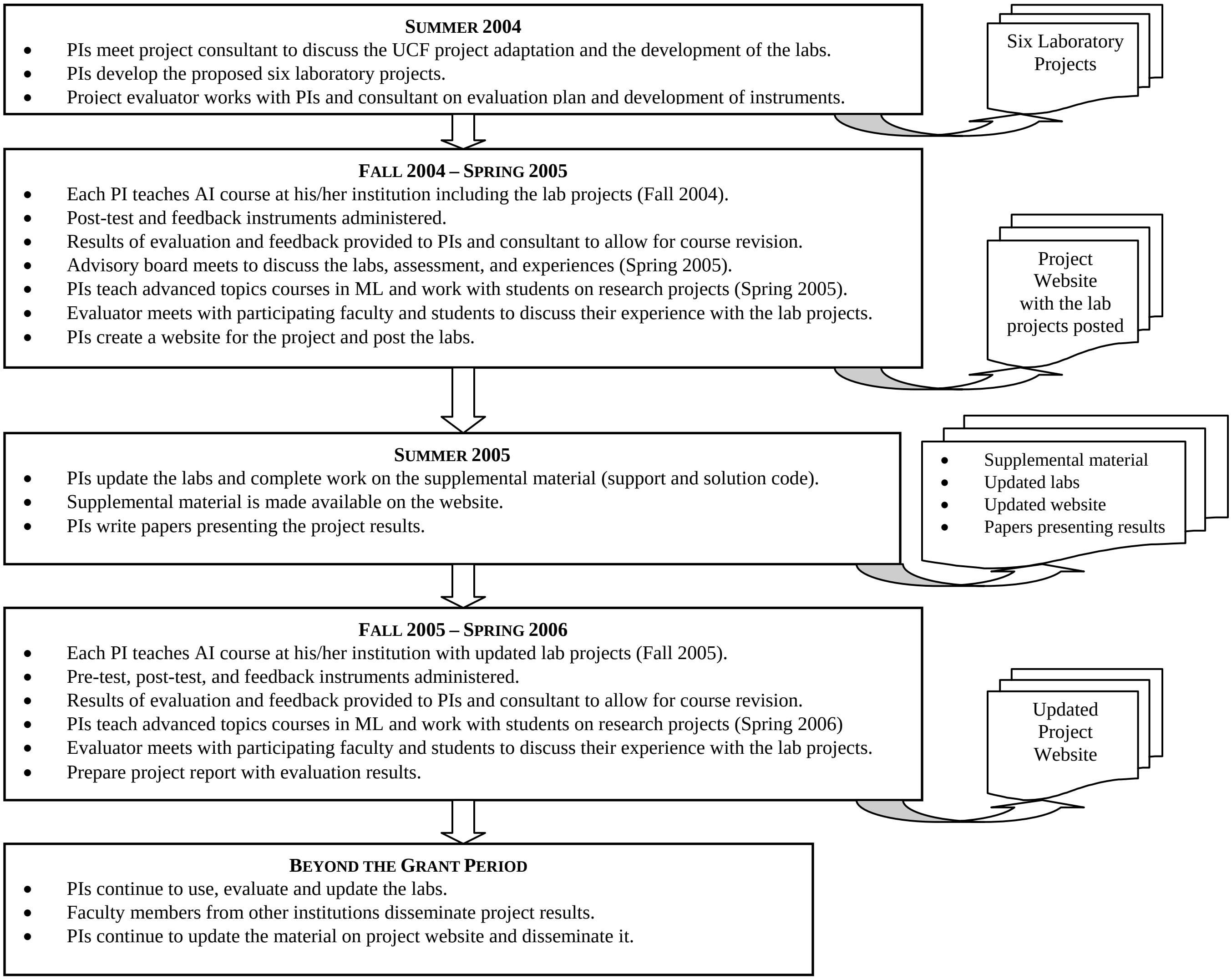
4. Overview of the Project

The project is geared toward the development of several intro AI projects, each of which involves the design and implementation of a learning system which will enhance a particular commonly-deployed application. Instructors may select which project(s) to assign throughout a semester or may give students options to select from.

The projects are easily adaptable. Our designs are modular and object-oriented, allowing instructors to customize assignments. While we envision using such a lab project for a full semester, the different entry points into the project will allow faculty to tailor these projects as they deem necessary for their course. At one extreme, the students may implement an entire machine learning system that illustrates core AI topics. At the other extreme, students may apply our solution code to understand the computational characteristics of the algorithms. In between is a range of choices allowing instructors to decide individually how much implementation is best for their students.

The material developed is being tested at the three PIs institutions, a small liberal arts college, a large state university, and a comprehensive private university. In addition, over 20 affiliate faculty will be testing some of the material in their courses and providing feedback.

Below is a project timeline with various stages in the development and implementation of the project including evaluation and dissemination plans.



SUMMER 2004

- PIs meet project consultant to discuss the UCF project adaptation and the development of the labs.
- PIs develop the proposed six laboratory projects.
- Project evaluator works with PIs and consultant on evaluation plan and development of instruments.

FALL 2004 – SPRING 2005

- Each PI teaches AI course at his/her institution including the lab projects (Fall 2004).
- Post-test and feedback instruments administered.
- Results of evaluation and feedback provided to PIs and consultant to allow for course revision.
- Advisory board meets to discuss the labs, assessment, and experiences (Spring 2005).
- PIs teach advanced topics courses in ML and work with students on research projects (Spring 2005).
- Evaluator meets with participating faculty and students to discuss their experience with the lab projects.
- PIs create a website for the project and post the labs.

SUMMER 2005

- PIs update the labs and complete work on the supplemental material (support and solution code).
- Supplemental material is made available on the website.
- PIs write papers presenting the project results.

FALL 2005 – SPRING 2006

- Each PI teaches AI course at his/her institution with updated lab projects (Fall 2005).
- Pre-test, post-test, and feedback instruments administered.
- Results of evaluation and feedback provided to PIs and consultant to allow for course revision.
- PIs teach advanced topics courses in ML and work with students on research projects (Spring 2006)
- Evaluator meets with participating faculty and students to discuss their experience with the lab projects.
- Prepare project report with evaluation results.

BEYOND THE GRANT PERIOD

- PIs continue to use, evaluate and update the labs.
- Faculty members from other institutions disseminate project results.
- PIs continue to update the material on project website and disseminate it.

Figure 2: Project Timeline

Six projects have been developed, each is briefly described below.

We have identified the following applications and learning models for our six lab projects:

Data Mining for Web User Profiling Using Decision Tree Learning: The project focuses on the use of decision tree learning to create models of web users.

Explanation-Based Learning and the N-Puzzle problem: The project involves the application of explanation-based learning to improve the performance of uninformed search algorithms when solving the N-puzzle problem.

Web Document Classification: The project investigates the process of tagging web pages using a topic directory structure and applies machine learning techniques for automatic tagging.

Character Recognition Using Neural Networks: The project involves the development of a character recognition system based on a neural network model.

Reinforcement Learning for the Jeopardy Dice Game “Pig”: In this project, students model the game and several illustrative variants, and implement various learning algorithms to compute optimal play, and experiment with such algorithms to experience their tradeoffs.

Getting a Clue with Boolean Satisfiability: We use SAT solvers to deduce card locations in the popular board game Clue, illustrating principles of knowledge representation and reasoning, including resolution theorem proving.

During the Fall 2004 semester, the Web Document Classification, Character Recognition Using Neural Networks, and Data Mining for Web User Profiling Using Decision Trees projects were classroom tested in the introductory AI course at the University of Hartford. At Gettysburg College, Reinforcement Learning for the Jeopardy Dice Game “Pig” and Getting a Clue with Boolean Satisfiability were tested in the corresponding AI course. At both institutions, the course is taught at the junior/senior level and requires the Data Structures course as a prerequisite. In the following sections we describe three of the projects in more detail followed by a discussion of each. Additional classroom testing will be done during the Spring 2005 semester. Preliminary evaluation results and our experiences follow. All projects developed along with sample syllabi will be made available at the project website at http://uhaweb.hartford.edu/ccli.

5. Description of Sample Lab Project: Explanation-Based Learning and the Eight Puzzle Problem

5.1 Overview

Typically a learning system uses domain knowledge and is expected to have some ability to solve problems. The objective of learning in this setting is to improve the system's knowledge or performance using that knowledge. This task could be seen as knowledge reformulation or theory revision. Explanation-Based Learning (EBL) uses a domain theory to construct an explanation of the training example, usually a proof that the example logically follows from the theory. Using this proof the system filters noise, selects the aspects of the domain theory relevant to the proof, and organizes the training data into a systematic structure. This makes the system more efficient in later attempts to deal with the same or similar examples. A classic AI problem, the N-puzzle problem, serves as a good application for illustrating this approach. In the 8-puzzle version, a 3×3 board consists of 8 tiles numbered 1 through 8 and an empty tile (marked as 0). One may move any tile into an orthogonally adjacent empty square, but may not move outside the board or diagonally. The problem is to find a sequence of moves that transforms an initial board configuration into a specified goal configuration.

The domain theory for the 8-puzzle problem can be expressed by a set of facts describing state transitions and a search engine that can be used to find paths between initial and goal states. Given a pair of an initial and a goal state (a training example), the search algorithm finds the shortest path between them (explanation or proof). Then applying the EBL techniques, the path is generalized so that it can be used later to match other initial states and bring the search algorithm directly to the goal state, without the resource-consuming exploration of the huge state space of the game. With carefully chosen training examples, useful rules for typical moves can be learned and then integrated into the search algorithm to achieve better performance.

The 8-puzzle problem provides an ideal setting for introducing conceptual AI search in an interesting and motivating way. The aim of the project is to investigate the effects of applying EBL techniques to improve the performance of search algorithms used to solve the 8-puzzle problem. The students benefit from this project in two ways: firstly, they learn a core ML technique and secondly, they better understand important issues related to computational complexity of uninformed and informed search and the role of heuristics in AI.

Students start with solving the 8-puzzle problem by using a set of standard search algorithms such as breadth-first, depth-first, depth-limited or iterative deepening. Then they apply informed search algorithms and investigate the role of heuristics to guide the search, compare performance, and try various approaches to find good heuristics. Next, students are introduced to the concepts of EBL and apply them to improve the performance of uninformed search algorithms when solving the N-puzzle. This is done in three steps:

1. Identifying useful search heuristics and generating and verifying the corresponding EBL training examples. Training examples are specified as pairs of start and finish game states.
2. EBL generalization step. In our setting, this step is basically substituting constants for variables and generating generalized state transition rules.
3. The last step in EBL is to add the new target concept definition to the domain theory. In the particular example, this means modifying the search algorithm to incorporate the new state transition rule.

Finally, students perform experiments with different training examples and measure the improvement in terms of run time and memory requirements. They also measure the effect of learning if too many examples or bad examples are supplied.

The students working on this project have usually already taken some major programming, algorithms, or data structure classes and have already been acquainted with the basic search algorithms as well as with their implementations in some major programming language such as C, Java or Lisp. For completing the project, they need implementations of these algorithms in any language. Preference however will be given to Lisp or Prolog implementations as they allow easier incorporation of generalized state transitions (EBL rules) by using the built-in mechanisms for pattern matching and unification that these languages provide. Lisp, Java, C, and other language implementations of search algorithms are widely available on the Web. A good collection of such algorithms is available from the companion web site of the popular AI textbook by Russell and Norvig[21] at http://aima.cs.berkeley.edu/.

Readings on search in AI (or problem solving by searching) are also widely available. Usually search is the first topic covered in AI texts. EBL is usually discussed in the ML sections of AI books in the context of the role of knowledge in learning (see section 19: Knowledge in Learning of Russell and Norvig[21]). The classical reference for EBL is Mitchell et al[12]. A project similar to the one discussed here, involving search and EBL in the domain of the 8-puzzle game is described in Russell et al[19].

5.2 Discussion

While enforcing core AI topics such as search, knowledge representation, and reasoning, the project allowed discussion of a variety of issues related to machine learning including:

- Better understanding of the concepts of learning and its relation to the area of search in AI; observation of how previous implementations can benefit from the use of machine learning.
- Performing experiments with training examples and measuring the improvement in terms of run time and memory requirements; measuring the impact on learning if too many bad examples are supplied.

6. Description of Sample Lab Project: Web Document Classification

6.1 Overview

Along with search engines, topic directories are the most popular sites on the Web. Topic directories organize web pages in a hierarchical structure (taxonomy or ontology) according to their content. The purpose of this structuring is twofold: first, it helps web searches focus on the relevant collection of Web documents. The ultimate goal here is to organize the entire web into a directory, where each web page has its place in the hierarchy and thus can be easily identified and accessed. The Open Directory Project (dmoz.org) and About.com are some of the best-known projects in this area. Furthermore, the topic directories can be used to classify web pages or associate them with known topics. This process is called tagging and can be used to extend the directories themselves. In fact, some well-known search portals such as Google return the relevant Open Directory topic path with the response, if applicable. As the Open Directory is created manually, it cannot capture all URLs, therefore just a fraction of all responses are tagged.

The aim of the project is to investigate the process of tagging web pages using a topic directory structure and applying machine learning techniques for automatic tagging. This would help in filtering out the responses of a search engine or ranking them according to their relevance to a topic specified by the user. Assuming that one knows the general topic of the web page in question, and that this is a topic in a topic directory, we can try to find the closest subtopic to the web page found. This is where machine learning comes into play. Using some text document classification techniques, one can classify the new web page to one of the existing topics. By using the collection of pages available under each topic as examples, one can create category descriptions (e.g. classification rules, or conditional probabilities). Then using these descriptions one can classify new web pages. Another approach would be the nearest neighbor approach, where by using some metric over text documents one finds the closest document and assigns its category to the new web page.

The project combines a number of important AI areas in a consistent way. Web search engines (1) use advanced search algorithms and information retrieval techniques to find web pages, and (2) use knowledge representation techniques to organize and structure the search results and create ontologies (i.e. topic directories).

The project is split into three major parts: data collection, feature extraction, and machine learning. These parts are also phases in the overall process of knowledge extraction from the web and classification of web documents (tagging). As this process is interactive and iterative in nature, the phases may be included in a loop structure that would allow each stage to be revisited so that some feedback from later stages can be used. The parts are well defined and can be developed separately (e.g. by different teams) and then put together as components in a semi-automated system or executed manually. Hereafter we describe the project phases in detail along with the deliverables that the students need to submit on completion of each stage.

Phase I consists of collecting a set of 100 web documents grouped by topic. These documents will serve as our training set. *Phase II* involves feature extraction and data preparation. During this phase the web documents will be represented by feature vectors, which in turn are used to form a training data set for the Machine Learning stage. *Phase III* is the machine learning phase. Machine learning algorithms are used to create models of the data sets. These models are used for two purposes. First, the accuracy of the initial topic structure is evaluated, and second, new web documents are classified into existing topics.

Three basic software components are needed to accomplish this project.

Search engine: Google or another search engine that uses topic directories and provides topic paths and searches within topics. The latter will be used to collect web pages as examples for the classification step.

Text processing software: This will be used to extract features from the web pages. Basically this is a text corpus analysis package that filters and extracts keywords with their frequency counts. These counts are then used to find the relevant subsets of features and to build the feature vectors describing the web pages[1].

Machine Learning software: This component is needed at the classification step when the web pages are converted into feature vectors. The Naïve Bayes and the nearest neighbor algorithms are used for classification (prediction of the web page topic), as these approaches have proven to be the most successful ones for text document classification.

6.2 Discussion

While enforcing traditional AI core topics, using a unified example, in this case web document classification, the project allowed the discussion of various issues related to machine learning including:

- The basic concepts and techniques of machine learning.
- Issues involved in the implementation of a learning system.
- The role of learning in improved performance and in allowing a system to adapt based on previous experiences.
- The important role data preparation and feature extraction plays in machine learning.
- The vector space model for representing web documents and a variety of feature extraction techniques combined with the pros and cons of each in identifying and classifying documents by feature vectors.
- The importance of model evaluation in machine learning and in particular the training and testing framework used to choose the best model for web page classification.

7. Description of Sample Lab project: The Game of Clue

7.1 Overview

The popular board game Clue (a.k.a. Cluedo) serves as a fun focus problem for this introduction to propositional knowledge representation and reasoning. After covering fundamentals of propositional logic, students first solve basic logic problems with and without the aid of a satisfiability solver (e.g. zChaff). Students then represent the basic knowledge of Clue in order to solve a Clue mystery. Several possible advanced projects are sketched if students wish to pursue the topic in more depth.

The murder mystery game Clue was first conceived in 1943 by the British law clerk Anthony E. Pratt while walking his beat as a wartime fire warden in Leeds. Since then, it has achieved the status of the world's most popular mystery game, and is one of the top-selling board games of the last half-century. Almost all students we have encountered have at least a passing familiarity with the game, and many have fond childhood memories of playing it.

Clue is primarily a *knowledge game* based on logical deduction, and thus provides an entertaining means of introducing fundamental concepts of knowledge representation and reasoning. The goal of the game is to be the first player to correctly name the contents of a case file: the murder suspect, the weapon used, and the room the murder took place in. There are 6 possible suspects, 6 possible weapons, and 9 possible rooms, each of which is pictured on a card. One card of each type is chosen randomly and placed in a "case file" envelope without being revealed to any player. All other cards are dealt out face-down to the players. Each player takes on the identity of one of the suspects.

Each player thus begins with the private knowledge that their dealt cards are not in the case file. As the game proceeds, players suggest possible suspect, weapon, and room combinations, and other players refute these suggestions by privately revealing such cards to the suggestor. This type of game is called a *knowledge game*, and the basic knowledge of the game may be expressed using propositional logic.

The atomic sentences used to express the knowledge of the game are of the form c_p, symbolizing the statement "The card *c* is in place *p*." For example, atomic sentence $\mathrm{pi}_{\mathrm{wh}}$ may symbolize the statement that "The player Mrs. **Wh**ite is holding the Lead **Pi**pe card." This statement may be true or false. Students quickly learn that knowledge representation is not a trivial exercise. Indeed most of the information gained over the course of a game of Clue has nothing to do with actually seeing cards. Most knowledge involves deductions concerning which cards other players do *not* have or, alternatively, cards privately shown between two other players. For success, students must gain a skill in thoroughly representing all relevant problem knowledge.

In order to provide warm-up exercises for knowledge representation and reasoning, we supply a number of simple logic problems which may be worked by hand. For example:

> *Suppose that liars always speak what is false, and truth-tellers always speak what is true. Further suppose that Ann, Bob, and Cal are each either a liar or truth-teller. Ann says, "Bob is a liar." Bob says, "Cal is a liar." Cal says, "Ann and Bob are liars." Which, if any, of these people are truth-tellers?*

This example problem is used to illustrate the propositional logic, conversion to conjunctive normal form (CNF), and resolution theorem proving. A set of eight classic propositional logic word problems are provided, ordered approximately by level of difficulty. A Java interface to a satisfiability (SAT) solver (e.g. zChaff) is provided, and students check their deductions using the SAT solver.

When students have achieved a level of comfort and competence with knowledge representation, their next task is to encode the knowledge gained over the course of a game of Clue in order to deduce whether or not any given card is in any given place (i.e. a player's hand or the case file). The product is an AI-assisted "detective notepad" program capable of guiding a player to expert deductions in the game.

In recent testing, such simple propositional reasoning outlined in the project clearly outperforms the "expert CPU" players in Hasbro Interactive's software *Clue: Murder at Boddy Mansion*, making deductions of the case file contents well before the Hasbro Interactive's AI. The same project software was also able to prove that Hasbro Interactive's AI should generally be able to correctly deduce the case file contents at earlier points in the game. Knowing that they are creating state-of-the-art AI adds excitement to the student's learning process.

At this point, an instructor could stop and be satisfied with a good propositional logic knowledge *representation* project. However, this is a natural point to begin an investigation of *reasoning* methods. At Gettysburg College, students then implemented the WalkSAT algorithm as a replacement for zChaff, and learned about other modern SAT solvers based on stochastic local search, including Novelty, Novelty+, and Adaptive Novelty+, the 2004 best SAT solver of the random category (http://www.satlive.org/SATCompetition/2004/). It was noted that the hand-coded adaptation heuristic for Adaptive Novelty+ could be finely tuned through the use of reinforcement learning techniques.

7.2 Discussion

The object of this project is to give students a deep, experiential understanding of propositional knowledge representation and reasoning through explanation, worked examples, and implementation exercises. Students:

- gain an understanding of the syntax and semantics of propositional logic, as well as general logic terminology, including "model", "(un)satisfiability", "entailment", "equivalence", "soundness", and "completeness".
- learn the process of knowledge base conversion to Conjunctive Normal Form (CNF).
- solve word problems with proof by contradiction (a.k.a. reductio ad absurdum) using resolution theorem proving.
- represent knowledge so as to complete a program implementation that performs expert reasoning for the game of Clue.

8. Evaluation Plan

The evaluation plan involves evaluating the effectiveness of the project in achieving each of the goals listed earlier through a multi-tier evaluation system involving (1) the students taking the AI course, (2) members of the advisory board, and (3) the three PIs on this project who are teaching the introductory AI course. Led by the project evaluator, the evaluation process involves both a formative evaluation, which will guide the development efforts, and a summative evaluation. An assessment of the project's effectiveness in improving student learning and an evaluation of our efforts at different stages of the project is used to assess progress made toward our goals and directs the development of our work. The project plan includes generation of instruments, data collection techniques, and data analysis. A sample student evaluation form is included in Appendix I.

Since on-going evaluation will be a systematic part of this project, adjustments to the material being developed takes place in light of the assessment results and as determined by feedback from students, faculty, and advisory board members. We have identified four faculty members who have a strong research record in the area of machine learning as well as in computer science education and who have extensive experience with curricular issues, including teaching the introductory AI course several times. These faculty members, along with two members from industry, serve on an advisory board that helps with the formative evaluation. The two members from industry were selected based on their area of expertise and on their experience in college level teaching in computer science and engineering. Dr. Michael Georgiopoulos of the University of Florida and the principal investigator on the NSF-CRCD project mentioned earlier serves as a consultant on this project, a member of the advisory board, and as an external evaluator of the project. He works with Dr. Susan Coleman on the evaluation of the project. The project timeline in Figure 1 includes the evaluation activities.

9. Our Experiences

We have completed the first phase of development and initial testing. Draft copies of the projects were developed during summer 2004. We worked with two students during the summer to do initial testing of the material before using them in a classroom setting. The fall 2004 semester was the first opportunity to class test the material. Five of the six projects were tested during the Fall 2004 semester by two of the PIs. The third PI is scheduled to teach the AI course during spring 2005 and will test additional projects. Modification based on student feedback is being incorporated. Additional testing will be done in the following semesters and further revisions will occur in response to testing.

The Web Document Classification, Character Recognition Using Neural Networks, and Data Mining for Web User Profiling Using Decision Trees projects were classroom tested at the University of Hartford. Students were given a choice of one of the three projects to do as a semester-long project. This is in addition to other smaller projects assigned throughout the semester. At Gettysburg College, students were asked to complete the Reinforcement Learning for the Jeopardy Dice Game "Pig" and Getting a Clue with Boolean Satisfiability projects.

Preliminary evaluation and feedback from students were very positive. Using a unified theme throughout the course proved to be helpful and motivating for the students. Students saw how simple search programs evolve into more interesting ones, and finally into a learning framework with interesting theoretical and practical properties. A copy of the questionnaire distributed to students at the end of the course is included in Appendix I. Results of selected questions from the questionnaire that was distributed to all students in the AI class at the University of Hartford are included in Figure 3 below.

Selected Questions from Evaluation Form	**Percentage of agree/strongly agree**
After taking this course, I have a good understanding of the fundamental concepts in Artificial Intelligence.	93%
After taking this course, I have a good understanding of the fundamental concepts in Machine Learning.	100%
The student project was an effective way to introduce Machine Learning concepts:	93%
Based on my experience with this course, I would like to learn more about Machine Learning and how it works.	77%
Based on my experience with this course, I would like to learn more about Artificial Intelligence and how it works.	85%
I am confident I can apply these AI problem solving techniques to different problems	93%
I had a positive learning experience in this course.	93%

Figure 3: Results of Selected Questions from the Evaluation Form at Hartford

These preliminary results show that our project goals appear to have been met, at least for this group of students. Student comments were generally very positive. They liked being able to apply techniques studied in class to real-world type applications. Further, they enjoyed being able to implement a system using the various techniques discussed in class while seeing how they all tie together. They also liked being able to see a working system where everything comes together in the end.

While this is a small sample to make conclusions from, these preliminary results are encouraging. Student results on test questions intended to test their understanding of the various

AI and machine learning concepts were consistent with these results. Further testing and a more thorough evaluation are being done by the project evaluator and a report is forthcoming as we continue to test and revise the material. In addition to the questionnaire, the evaluation process includes interviews with students and faculty.

Conclusion

We presented our experiences with a project in progress, funded by NSF, involving the use of machine learning as a theme to teach AI concepts. Preliminary results were very positive and showed that students had good experiences in the classes. Several projects that students worked on throughout the course were presented. The lab projects presented are cost-effective in that all software needs and tools are available at no cost while the hardware need is restricted to a general purpose computer laboratory that is available at most computer science departments. This cost-effective teaching model may be easily replicated at other institutions large and small, public and private.

Overall, student experiences were very positive. While covering the main AI topics, the course provided students with an introduction to and an appreciation of an increasingly important area in AI, Machine Learning. Using a unified theme proved to be helpful and motivating for the students. Students saw how simple search programs evolve into more interesting ones, and finally into a learning framework with interesting theoretical and practical properties. Based on this first year experience, the projects will be revised accordingly and made available on the project website for further testing next year as well as for testing by affiliate faculty and other interested parties. We will use the revised projects again next year and will continue to revise based on our experience as well as the experience of others. A full evaluation of the project will also be done as per the project timeline of Figure 1.

Acknowledgement

This work is supported in part by National Science Foundation grant DUE CCLI-A&I Award Number 0409497.

References

1. Chakrabarti, S., *Mining the Web - Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, 2002.

2. Engel, G., and Roberts, E., Eds., *Computing Curricula 2001 Computer Science: final report,* December 15, 2001, Los Alamitos, CA: IEEE Press, 2002.

3. Fox, S., *Artificial Intelligence and Robotics Lab*, http://www.macalester.edu/research/fox/, 2000.

4. Georgiopoulos M., Russell I., Castro J., Wu A., Kysilka M., DeMara R., Gonzalez A., Gelenbe E., Mollaghasemi, “A CRCD Experience: Integrating Machine Learning Concepts into Introductory Engineering and Science Programming Courses”, *Proceedings of the 2003 ASEE Conference*, June 2003.

5. Ghani, R., *CMU Web->KB project*, http://www-2.cs.cmu.edu/~webkb/, January 2001

6. Hearst, M., Improving Instruction of Introductory Artificial Intelligence, AAAI Press, 1995.

7. Kumar, A., "Using Robots in an Undergraduate Artificial Intelligence Course: An Experience Report", *Proceedings of Frontiers in Education Conference*, 2001.

8. Kumar, D., and Meeden, L., "A Robot Laboratory for Teaching Artificial Intelligence", *Proceedings of SIGCSE*, pp. 341-344, February 1998.

9. Levine, D., Harlan, R. and McClarigan, S., "The Khepera Robot and the kRobot Class: A Platform for Introducing Robotics in the Undergraduate Curriculum", *Proceedings of the ACM Technical Symposium on Computer Science Education*, February 2001, pp. 105-109.

10. Lundeberg, M. A., and Diemert, S. (1995). Influence of Social Interaction on Cognition: Connected Learning in Science. *Journal of Higher Education*, 66(3), 312-335.

11. Markov, Z., *Machine Learning*, http://www.cs.ccsu.edu/~markov/ccsu_courses/MachineLearning.html

12. Mitchell, T.M., *Machine Learning*, McGraw Hill, New York, 1997.

13. Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, S.T., *Explanation-Based Generalization: A Unifying View, Machine Learning 1*, Kluwer, Boston, 1986, 47-80.

14. Mladenic, D., *Yahoo Planet project*, http://www-ai.ijs.si/DunjaMladenic/yplanet.html, June 2001.

15. Neller, T., *Machine Learning*, http://cs.gettysburg.edu/~tneller/cs391/, February 2003.

16. Nilsson, N., *Artificial Intelligence: A new Synthesis*, Morgan Kaufmann, Publishers, 1998.

17. Russell, I., Georgiopoulos M., Castro J., Neller T., McCracken D., King L., Bouvier D., "Condensing the CC-2001 Core in an Integrated Curriculum", *Journal of Computing Sciences in Colleges*, 18(5), 65-69, 2003

18. Russell, I., "A Neural Network Simulation Project", *Journal of Artificial Intelligence in Education*, 3(1), 1992.

19. Russell, I., Z., Markov and N., Zlatareva. "Introducing Machine Learning from an AI Perspective", *Proceedings of the 13th International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP 2003)*, 469-473.

20. Russell, I. and Neller, T., *Implementing the Intelligent Systems Knowledge Units of Computing Curricula 2001*, Proceedings of ASEE/IEEE Frontiers in Education Conference, November 2003.

21. Russell, S. J. and Norvig, P., *Artificial Intelligence: a modern approach*, Upper Saddle River, NJ: Prentice-Hall, Second edition, 2002.

22. Witten, I. H. and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.

Biographical Information

INGRID RUSSELL is a Professor of Computer Science at the University of Hartford. Her research interests are in the areas of artificial neural networks, pattern recognition, and computer science education. Her work has been supported by grants from NSF, NASA and the Connecticut Space Grant Consortium. She serves on the editorial board of the International Journal of Intelligent Systems and has served in several editorial capacities.

ZDRAVKO MARKOV is an Associate Professor of Computer Science at Central Connecticut State University. His research interests are in the areas of Machine Learning, Data and Web Mining, and Logic Programming. He has published 3 textbooks and more than 40 conference and journal papers in these areas. He has served on a number of program committees of AI conferences.

TODD W. NELLER is an Assistant Professor of Computer Science at Gettysburg College. He received his Ph.D. with distinction in teaching at Stanford University, where he was awarded a Stanford University Lieberman Fellowship, and the George E. Forsythe Memorial Award for excellence in teaching. Neller's current research involves applications of reinforcement learning to the control of combinatorial optimization and search algorithms.

MICHAEL GEORGIOPOULOS is a Professor of the School of Electrical Engineering and Computer Science at the University of Central Florida. His research interests lie in the areas of neural networks and applications of neural networks in pattern recognition, image processing, smart antennas and data-mining. He is an Associate Editor of the IEEE Transactions on Neural Networks since 2001.

SUSAN COLEMAN is the Ansley Professor of Finance at the University of Hartford. Her research interests include small firm capital structure as well as research on women-owned and minority-owned small firms. Coleman served as the Internal Evaluator for the University of Hartford's College of Engineering NSF grant "Integrating Engineering Design with the Humanities, Social Sciences, Sciences, and Mathematics" from 2001 through 2002.

Appendix I
Evaluation Questionnaire

Introduction to Artificial Intelligence
Evaluation of Student Project
Fall 2004

Gender:

1	2
Male	Female

Year:

1	2	3	4
Freshman	Sophomore	Junior	Senior

Questions 1 – 22 are answered on the following scale:

1	2	3	4	5
Strongly Disagree	Disagree Somewhat	Neither Agree nor Disagree	Agree Somewhat	Strongly Agree

1. It was easy to access the student project for the course.

2. Requirements for the student project were clearly presented and instructions were easy to follow.

3. It was easy to contact the professor with questions about the student project if there was something I did not understand.

4. The time allowed for completion of the student project was sufficient.

5. The student project was interesting to work on.

6. The student project contributed to my overall understanding of the material in the course.

7. I liked working on a team to do the student project.

8. I feel that I learned more by working on a team than I would have if I had done the project by myself.

9. The student project took a reasonable amount of time to complete.

10. The student project was at an appropriate level of difficulty given my knowledge of computer science and programming.

11. The feedback that I received on the student project was clear and easy to understand

12. After taking this course I feel that I have a good understanding of the fundamental concepts in Artificial Intelligence.

13. After taking this course I feel that I have a good understanding of the fundamental concepts in Machine Learning.

14. The student project was an effective way to introduce Machine Learning concepts.

15. Based on my experience with this course, I would like to learn more about Machine Learning and how it works.

16. Based on my experience with this course, I would like to learn more about the field of Artificial Intelligence.

17. The Artificial Intelligence problem solving techniques covered in this course are valuable.

18. I have a firm grasp of the problem solving techniques covered in this course.

19. I would like the opportunity to apply some of these problem solving techniques in the future.

20. I am confident that I can identify opportunities to apply these problem solving techniques.

21. I am confident that I can apply these problem solving techniques to different problems.

22. I had a positive learning experience in this course.

Essay questions:

23. What I liked best about the students project:

24. What I liked least about the student project: