Web-based Artificial Intelligence Agents

Sandip C. Patel, Erastus Karanja, Suneel K. Maheshwari

ABSTRACT

This paper discusses implementations of Artificial Intelligence (AI) technologies over the World Wide Web. The implementations include a pilot agent with natural language processing and rule-based AI that recommends IBM products, a planning agent that can be used by dispersed Army Small Units, and a learning apprentice for the Web that can help users to locate desired Web sites. The potential for the use of the AI technology, in light of the latest development in Internet and Web 2.0 technologies, is high for both corporations and every day users.

Keywords: Artificial Intelligence, Intelligent Agents, Natural Language Agents, Agents on the Internet

INTRODUCTION

One of the features on Apple's new iPhone 4S is a voice activated AI program called Siri, which is marketed as a modern "personal assistant"(Claburn, 2011). The features of this program closely resemble those on IBM's Watson which won the TV show "Jeopardy"(Paul, 2011) by answering the questions more correctly than its two human opponents. Both systems are a result of profound advancements in AI, which allow users to give voice commands and receive answers to questions in real time and, thereby, revolutionize the day-to-day human-machine interactions. AI systems achieve this by increasing the speed at which data and information can be accessed and retrieved. The recent advances in e-commerce and Web 2.0 technologies have resulted in the production of a lot of data and information. To the vendors and customers, these colossal data and information repositories can lead to information overload. An AI system is helpful in quickly delving through the data and retrieving relevant information.

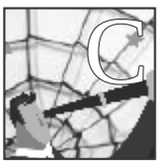
AI is emerging as one of the methods for accessing and managing data and information. AI also helps in creating knowledge, and it has various useful practical applications. For instance, AI is applicable to e-commerce because of its capability to reduce search costs (Hinz& Eckert, 2010).With the US wireless penetration reaching 96% (CTIA, 2011) and almost 70% of US households having access to computers and the Internet (US Census Bureau, 2009), the future of e-commerce is certain. Thus, successful information access and retrieval is of paramount importance to both commercial users/vendors and customers.

An AI-based search engine can be an asset to firms that want to provide an effective tool by which their customers can find exact products or information-match that tally with their requirements. On the other hand, non-commercial users, such as government agencies, can take the advantage of combining the AI technology with the Web. For example, by using a planner with AI that works on the Web, the dispersed army units can collaboratively work on a strategic plan and, thus, benefit from the specialized skills and knowledge of dispersed team members (Martins, Gilson, & Maynard, 2004).A household user can also benefit by using the AI technologies. For instance, a user surfing on the Internet in search of a subject matter with any general interest will probably not have specific knowledge about the contents of the collections on the Internet. The AI agent can help such a user to navigate through the jungle of hyperlinks by coalescing and organizing the data and information into a more manageable and understandable format. In essence, the AI technology has the potential of being an indispensable part of the day-to-day human-machine interactions on the Internet.

This paper discusses three AI agents which implement different AI technologies in spite of being web-based. The first agent, called Natural Language Assistant comprises a search engine with natural language processing capability that is based on the traditional rule-based AI Technology. This implementation is timely considering the demand of the AI implementation in the search technology. For example, Google[1], even with its highly successful search engine, acknowledges the limitations of a traditional search engine and argues[2] that the ultimate search engine must have an AI component to it.

The second AI agent covered in this paper shows how an AI planning technique and capability can be exploited as an aid to support US Army Small Unit operations. A web-based AI planning agent named O-Plan has been developed for use during the planning life cycle in military and civilian crisis situations. The paper also covers a third agent that explores the notion of a tour guide software agent for assisting users browsing the Internet. A Web tour guide-agent provides assistance similar to that provided by a human tour guide in a museum. The agent guides the user along an appropriate path through the collection, based on the agent's knowledge of the user's interests, location, and relevance of the various items in the collection. The agent also incorporates knowledge generated from past experiences of the way other users have interacted with the collection in the past.

The rest of the paper is organized as follows. We begin by exploring the features of the three AI agents as well as their strengths and weaknesses. Following that, we discuss the implementation issues related to the three identified AI agents. A comprehensive comparison of the three AI agents is followed by concluding remarks.



COMMON TYPES OF ARTIFICIAL INTELLIGENT AGENTS

Natural Language Assistant

The Natural Language Assistant (NLA) is a web-based natural language dialog system that helps users to find relevant products on e-commerce sites (Chai et. al. 2002). There are a number of popular implementations of AI agents that are based on NLA technology. The version of NLA described in Chai et al. (2002) was developed as a pilot agent that recommends IBM ThinkPad laptops based on the customers' preferences. The front end (user interface module) accepts both type-in text and speech input with the parser capturing the user's specifications (e.g. speed, memory, usage pattern, and price).Based on these expressions and the session context maintained by the dialog manager, the interpreter constructs a set of constraints on attributes of IBM Think Pads. These constraints are then translated into an SQL query by the action manager. The action manager then executes the SQL query against a relational database and eventually retrieves a set of products matching user specifications. Based on the identified constraints and the retrieved output, the AI agent retrieves the products online, and gives its response which can be further questions, recommendations, or both. This output is presented through the user interface module under which the user can start another interaction with the agent if need be.

[1] with seven consecutive quarters of profit earnings

[2] Source: PBS interview of a Google executive, 11/30/2002.

Smart Choice is a nNLA based AI agent that is implemented as an online content-based recommender system. It provides parents with information about schools, based on the parents' preferences and the needs, interests, abilities, and talents of the student(Wilson et al. 2009). Closely resembling Smart Choice is another NLA based AI agent called Auto Tutor (Graesser, Jeon, &Dufty, 2008; Graesser, Chipman, Hayes, & Olney, 2005). Auto Tutor generates deep reasoning questions, and prompts students for answers, while also offering pointers through dialogue. The dialogue involves, for instance, requesting clarifications as well as soliciting for more information from the student in order to narrow the choices and enable the learner to deduce the correct answer.

There are many implementations and applications of NLA based AI agents to the general public; hence, the three agents mentioned here are mere examples. The following table presents the strengths and weaknesses of the NLA based AI agents discussed above.

STRENGTHS
NLA based AI agent is a generic concept that can be easily applied to many e-commerce websites for product recommendations as well as online services that incorporate question and answer sessions.
Unlike menu-driven navigation or the keyword search engines that are based on word matching, NLA based AI agents work by extracting semantics, a concept that reduces user frustrations.
NLA based AI agents offer more flexibility through a number of methods: a) User interface: the system allows and provides both text and speech input/output b) User/system interaction: while answering the questions from the system, the user is not confined to specific answers. c) Context: the user has leeway in specifying the requirements and can utilize natural-language relational operators (e.g. for a ThinkPad recommender, between $500 and $800 and with at least 3 hours of battery life) without being confined to a pre-specified selection set.
The fact that the parser uses a statistical approach makes NLA-based AI agents such as Smart Choice and Auto Tutor easily scalable to other languages. The existing schema can be used to annotate a collection of sentences from another language.
WEAKNESSES
For a novice user who just needs a quick answer, the NLA-based AI agents systems can be overwhelming.
The NLA-based AI agents would be more appropriate to both novice and experienced users if it incorporated menu driven interface in addition to the already implemented text and speech recognition.
The issue of response time is not clearly addressed in the literaturealthough it might not be a concern because of the advancement in processor and memory technology. Nevertheless, with centralized systems and many user interactions, real time applications that generate multiple messages parsingcan lead to bottlenecks.

O-Plan

O-Plan is based on a command, planning, and control architecture (Tate, Levine, Jarvis & Dalton, 2000; Tate, Dalton, & Levine 2000; Tate, Dalton, & Levine 1998; Tate 1997a;Tate 1997b;Tate 1997c;Drabble& Tate 1995; Fraser & Tate 1995). Since O-Plan is domain independent, it is intended to provide flexibility through an open modular structure and has been implemented on the Web. The O-Plan project is undertaken at the University of Edinburgh and is part of DARPA/Roma Laboratory Planning Initiative. This collaborative venture aims at studying mixed initiative planning methods, and how these methods can be applied to realistic problems in logistics, air campaign planning, and crisis action response.

The model for the O-Plan agent is based on planning workflow control. The implementation employs an underlying representation of plans termed <I-N-OVA>, (Tate, 1995) which expresses plans as a set of constraints on behavior. O-Plan can help with planning and is easily applicable to, for instance, disaster relief planning—where a number of people need to be evacuated from various places under different circumstances. In such an incident, for example, stormy weather will prevent helicopters from being used, while collapsed bridges will impede road rescues. O-Plan generates a suitable plan of action and helps during its execution. The agent can also be used by the military in the deployment of surveillance aircraft, thus helping the US troops to, for instance, hunt down their enemies(Erwin, 2008).

Prior to carrying out a mission[3], an Army Small Unit is provided with an operation order which is prepared at a higher echelon. Operation orders are usually written in English and follow a strict structure but they are shared between military planners and planning assistants. The operation orders originate from initial requirements against which company commanders can establish options for the course of action to be taken. O-Plan is applicable during these initial stages when there is room to explore various points of views, assumptions, advice on tactics, etc. O-Plan is also preferred in latter stages and can be used by multiple users performing different tasks (for instance, monitoring the plan execution).The users can interface to both O-Plan and each other via a panel that is accessible through the various Web browsers. This panel assists in coordination of multiple participants.Lopez et al. (2004) point out that O-Plan is expected to help the United

[3] Actual details acquired by the authors from US Army Small Unit Operation's Subject Matter Experts (SMEs) at Fort Benning, Georgia.

States adopt better strategies to combat the enemy following the attacks of September 11, 2001. Thus, O-Plan is intended to be a proactive and a strategic planning tool to avert man-made disasters and also recover from natural disasters.

The Planning Phase of O-Plan

The planning phase of O-Plan entails, among others, the formulation of objectives by the user where by the operation orders (requirements) serve as the inputs. A user-friendly simple pull-down menu is used to describe the plan objectives, possible resources, time constraints, and operational approaches. O-Plan offers a mixed initiative mode, which ensures that all users with different authorities can seamlessly interact with each other. To implement this capability, the planner is authorized to generate and explore any of a number of sub-options for the top level plan and interact with various users with different degrees of authority. For instance, top level executives such as a commander have the capability to access all the available operation options, plans, and approaches and can also generate the action evaluation matrix.

Following the generation of a set of plans, the most appropriate plan is selected for execution. During the execution phase, O-Plan provides a "goal structure" which offers casual links indicating the purpose of actions in the plans, and how these plans are interrelated. An action may trigger a situation that precedes another action. For example, in a hostage rescue plan, a combat unit should first secure its toehold building before proceeding to the building that houses the hostages. The precondition actions will be specified in the goal structure. In addition to helping with determining the preconditions required for completing an action, O-Plan also helps in re-planning after a break in prior conditions or change in circumstances.

Moreover, O-Plan works as a re-planning agent by generating new plans after a break in preconditions. On the other hand, when the break occurs in the middle of a plan, O-Plan can generate a report that shows the actions that have already been completed while also adding new actions to the remaining part of the plan. The user input and interaction in any of these stages is similar to those employed when constructing a new plan.

The authors participated in the O-Plan through links:http://www.aiai.ed.ac.uk/~oplan/isd/andhttp://www.aiai.ed.ac.uk/project/oplan/isd/. Below are some of the strengths and weaknesses of this AI agent.

STRENGTHS
O-Plan offers interactive demonstrations which are available at
http://www.aiai.ed.ac.uk/~oplan/isd/andhttp://www.aiai.ed.ac.uk/project/oplan/isd/
By taking the user through most envisioned planning phases through a user-friendly interface, the system provides thoroughness of the support for planning.

WEAKNESSES
Although the interactive demonstrations mentioned earlier were insightful, some fields were not clear enough as to what was expected as input.
The literature (Tate, Levine, Jarvis, & Dalton, 2000; Tate, Dalton, & Levine, 2000; Tate, Dalton, & Levine, 1998; Tate, 1997a;Tate, 1997b; Tate, 1997c;Drabble & Tate, 1995; Fraser & Tate, 1995)fails to provide evidence of guaranteed behavior of O-Plan on the Web.
There was also no literature that showed how the O-Plan would respond with many planners working simultaneously and not just sharing it on the Web.
Similarly, the existing literature failed to offer information on how issues of access control and web security are/will be addressed.

Web Watcher

Web Watcher (Armstrong, Freitag, Joachims, & Mitchell, 1995; Joachims, Freitag, &Mitchell, 1997; Joachims, Freitag, &Mitchell, 1996;Joachims, Mitchell, Freitag, & Armstrong, 1995) is an AI agent that interactively helps users to locate desired information on the Internet. It makes use of learned information that is based on the user's specified goals and previously visited hyperlinks to predict which hyperlinks are more likely to lead to the desired target information for the user.

Users invoke Web Watcher by following a Web hyperlink to its Web page. After that, a form is generated on which the user inputs information that specifies what information is sought. Following this, the Web Watcher retracts and points the user back to the initial page where the user was visiting before accessing the Web Watcher website. Then, Web Watcher follows the user's actions in what is referred to as "looking over the user's shoulder" (Armstrong, Freitag, Joachims,& Mitchell, 1995). Web Watcher offers suggestions for hyperlinks by highlighting these hyperlinks on the current page, and adding new hyperlinks while suggesting others that are closely related to the current page. Web Watcher also has the capability to send an email message to the user, at a later pre-specified time, when a particular page has changed.

We tried using Web Watcher which was supposed to be accessed through http://www-2.cs.cmu.edu/~webwatcher/ but found that the support had been discontinued. We took the demo instead.

In Joachims, Mitchell, Freitag, & Armstrong (1995), work in progress was described. The authors implemented an algorithm using the "nearest neighbor" approach to map an arbitrary Web page to a set of related pages. A matrix representing related pages as rows and hyperlinks as columns was used to "relate" the pages with one another. The "nearness" to a page was measured by the Term Frequency/Inverse Document Frequency (TFIDF) vector described below. Each page had two "neighbors" which had

the closest distance to it. Each hyperlink a was described by the TFIDF vector representation of the underlined anchor text. Each page s was represented in an analogous way using the text in its title. For the purpose of determining similarity, the distance between the hyperlink a1 on page s1 and the hyperlink a2 on page s2 was defined to be the distance between a1 and a2 plus twice the distance between s1 and s2. The distance between two vectors was defined as the cosine of the angle between the vectors, a standard similarity measure used in TFIDF. The target function learned using the AI was a mapping from an arbitrary Web page to a set of related pages. The authors mentioned other alternative algorithms such as mutual information (Quinlan, 1993)and Minimum Description Length (Rissanen, 1978). Instead of describing the algorithm implemented in Joachims, Mitchell, Freitag, and Armstrong (1995) in more detail, we will do so for the later version of Web Watcher in the following paragraphs.

In Joachims, Freitag, and Mitchell (1996) the authors use a different matrix which is based on a technique from information retrieval. User interests and hyperlink descriptions are represented by vectors, with each dimension representing a particular word in the English language. Using the TFIDF heuristic, the weight of a word w for a piece of text is computed as:

$$TFIDF(w,d) = TF(w,d) * \log(n/DF(w))$$

The term frequency TF(w,d) counts the number of times word w occurs in text d. The document-frequency DF(w) is the number of texts that contain word w and n is the total number of texts. The TFIDF measure assigns a word a higher weight if it occurs more often in a piece of text. The words that occur more frequently throughout all texts receive a lower weight. The weight of a word is zero if it does not occur in a piece of text at all. Based on this vector representation, the similarity of two pieces of text can be calculated as the cosine between their vectors.

The algorithm that suggests hyperlinks goes through all the hyperlinks on the current page. Using the TFIDF method to find the similar pages, each hyperlink is ranked using its similarity value. The hyperlinks that are sufficiently similar (above a threshold) to the user's interest are suggested by Web Watcher.

Close to six thousand tours were given[4] between August 1995 and March 1996. Web Watcher offered at least one piece of advice in 21% of the cases, and this was demonstrated by the highlighted hyperlinks. However, if the users had randomly followed the pages while ignoring the advice of Web Watcher, it still would have offered advice 15% of the times. As such, the results indicate that the effective "learning" took place only 6% of the time, which is not very encouraging. As we observed in Armstrong, Freitag, Joachims, & Mitchell, 1995; Joachims, Freitag, Mitchell, 1997; Joachims, Freitag, Mitchell, 1996; and Joachims, Mitchell, Freitag, & Armstrong, 1995, the authors were trying to find better AI techniques. The experiment also concludes that in about 44% of the cases, the user followed the advice given by Web Watcher. Our concern with this result was that Web Watcher did not indicate whether the users were satisfied with the results they received in those 44% cases.

[4]as per http://www-2.cs.cmu.edu/~webwatcher/ accessed on 10/02/2010

Practical Applications of Web Watcher

There are a number of practical applications of the Web Watcher and we will briefly discuss some of them here. One in particular was presented by Torres-Verdin et al. (2004), and is an intelligent system that takes in a set of queries and attempts to learn the search profile of a specific user. Thus, the system aims at improving the correlation between the user's search interest and the retrieved documents. The system also seeks to help users retrieve and organize information which is available across huge data and information repositories (Deltor, 2002).

Another Web Watcher agent called Expert Clerk (Shimazu, 2002) imitates a salesclerk and consolidates the requests from a human shopper. It utilizes gentropy (navigation by asking) and also shows contrasting sample products which are accompanied by explanations of their selling points (navigation by proposing) to narrow down the list of products. Web Crow is another implementation of a Web Watcher which is used as a crossword solver and employs knowledge mined from the Web to solve the puzzles. It makes use of the Web and natural language processing techniques in order to generate clues to solve the puzzles. Web Crow has utilized the web-based approach to outperform human challengers in solving crossword puzzles (Ernandes, Angelini, &Gori 2008).Finally, Twine, a Web organizer, is based on Semantic Web technology and processes the meaning of Web pages in lieu of downloading them. Twine also has features that allow users to track and organize relevant information about products, people, places, items, documents, and recipes based on the user's interests (Farber, 2008).

I IMPLEMENTATIONS

We have incorporated some details on the use of AI in each agent in the above sections. This section exclusively addresses the issue of the AI implemented, including the discussions using our analysis.

NLA

NLA's web-based AI agent is implemented by two of its components. The "interpreter" component extracts constraints that specify relations and values for product attributes (for example, shipping-cost<$30). Constraints may be either numeric or string. Numerical constraints are normalized to canonical units, such as dollars for "shipping-cost." String-valued attributes are either matched directly (using the substring matching) against a string value in the product database, or used as a similarity measure to produce canonical string values. The similarity measure technique uses a particular attribute category with values for that attribute in the product database. For example, in the query, "I want a 24-inch monitor," the interpreter identifies a constraint "size=24" because the interpreter has various values for the size attribute. The "action manager" component simply translates constraints to SQL code-lines. To translate the

multiple min-max constraints, it reverses the order of occurrence. For example, for the query "cheapest, latest laptop with 2 GHz", the action manager first searches for 2 GHz, then the latest, and finally the cheapest.

The above-discussed are the main AI components. The interpreter implementation seems traditional and straight forward. We verified that the interpreter allows multiple constraints on the same attribute. However, the action manager is too simplistic. We deem that, with the limited database for this pilot agent, it may be justified to have a limited-functionality action-manager. Nonetheless, for a real application, users need to implement a dedicated constraint solver for better speed. Considering goal satisfaction, grounding techniques, and unification methods used by various constraint solvers, users can select an appropriate constraint solver. Users will have to tackle the issues of coming up with the criteria for pruning. For example, if a customer asks for a price of less than $800, then the part of the decision tree representing the laptops with 2GHz and higher processor speeds can be pruned since such classes of laptops will not fit the user's dollar criteria. Also, to translate the multiple min-max constraints, the interpreter could probably use a suspended goal strategy rather than simply reversing the order of occurrence of constraints. This probably would not result in correct interpretation all the time. The authors have not mentioned about how the disjunction (or) is handled. This limitation may result from the fact that the interpreter does not understand a disjunctive criterion.

In general, we feel that NLA makes an appropriate but limited use of AI. Musliner et al., 1995 have found that "the broad application of AI methods to real-time domains will require new approaches, differing from many of the traditional search-based techniques explored in the field. Based on our testing, we feel that the pilot agents such as NLA will need to go with the in-depth AI implementation for their use in the "real world".

O-Plan

We have extracted the AI implementation information for O-Plan, which can be categorized as following:

- Least-commitment approach: Under this inference procedure, the constraint satisfaction algorithms do not commit to a constraint; instead, they delay making the choice. This approach is selected most probably because O-Plan implements partial order planning and the least commitment approach forms the basis for this category of planning.
- Partial-order planner: Such a plan can take an advantage of not having to commit to an ordering between tasks until the application is forced to do so. While building a partial order plan between actions, performing an action twice can create a problem. An algorithm would require assigning an index to each instance of an action (not the action itself) and then order the instances. O-Plan implementation details are sketchy and did not list the exact algorithm if users used this technique to guard against the problem.
- Opportunistic selection: In case there are multiple actions satisfying the constraints, the strategy needs to be chosen to deal with the constraints. The opportunistic selection chooses the action with the highest utility upper bound. The literature does not mention the justification for picking this selection criterion over the two other selection rules, which areselecting the average highest utility and highest utility lower bound. A statistical analysis is usually carried out to find the most appropriate criteria, but, in general, when the greatest reward potential is high, this analysis is necessary. It can be deduced that the O-Plan was probably developed to maximize the benefit (the highest utility possible) from an action.
- Localized search to explore alternative actions: We investigated to see if O-Plan attempts to reach the global maxima. We found that it does indeed use "global re-orientation" alternative when necessary.

Since a plan can be viewed as a set of constraints which limit the behavior when it is executed, we looked into the types of constraints used by O-Plan. It uses the following three types:

(1) Implied constraints: The pending constraints that get added when unsatisfied requirements get handled.

(2) Plan node constraints: The nodes can be actions in an activity planner or resources in a resource planner.

(3) Detailed constraints: These can be temporal constraints—which treat time as any other resource, or they can be variable constraints—which put restrictions on plan objects. The advantage of the temporal constraint is that if the goal state specifies a deadline and if the partial plan requires more time than allowed, we can backtrack immediately without considering any completion of the plan. This advantage results from the fact that time never goes backward.

After testing the AI planners and examining the details on O-Plan planning constraints, ,Although AI planners can implement given constraints, we contemplated the question on whether the planners can replace humans who can discern between two plans, or relax criteria in order to explore tradeoffs. We were glad to find that theO-Plan Project considered combining the O-Plan planner with a knowledge-based system that reasons about the plan evaluation.

WebWatcher

The algorithm that suggests the related hyperlinks to the user uses a function named User Choice?, which takes the following form:

User Choice?: Page x Interest x Link

Where Page is the current page, Interest is calculated using the TFIDF method described earlier and the Link is one of the hyperlinks found on the Page. The user's interests are matched against the hyperlinks. Each time the user selects a new hyperlink, a training example is logged for each hyperlink on the current page corresponding to the Page, Interest and Link. Web Watcher splits this function for each Web page. (In other

words, they have different input vectors for each page). So, Web Watcher learns a function User Choice? p for every page p as demonstrated in:

User Choice? p: Interest x Link

Although the authors did not acknowledge the inadequacy of this learning method explicitly, from the experiment results described in the section on Web Watcher, we think that there is a scope for improvement in the current AI technique used for learning. The authors have dedicated large portions of literature (Freitag, and Mitchell, 1996), which explore alternative learning methods. One of the two alternative methods is to employ reinforcement learning so as to create more refined descriptions of the hyperlinks. The other idea combines the results of multiple learning methods in order to improve accuracy. We will discuss only the former, taking into account the scope of this survey in the following paragraphs.

The "goodness" of an action is expressed in terms of an evaluation function defined for all possible state-action pairs. An analogous navigation problem for an agent on the Web is to navigate from page to page by taking hyperlinks so as to get to good sources of information. Pages can correspond to states, and hyperlinks can correspond to actions. The reward that a Web agent receives for a page is the degree to which the result fits the user's interests. The learning agent then seeks for tours through the Web that leads as directly as possible to pages of greatest interest to the user. The reward function that measures whether a page fits the user's interest will be based on TFIDF heuristic that was earlier discussed.

A significant problem with the above approach might be that the users do not always stick to pages they have already seen. We classify this environment as semi-dynamic. In a semi-dynamic environment, it is difficult for a system to "learn" effectively; so, the nearest neighbor approach could be an approximation at best.

COMPARISON OF THE AI AGENTS

This section compares and contrasts the three agents that we examined earlier. Where applicable, we arranged the agents starting from the highest/best to the lowest/worst in their respective categories. For example, we listed the product with the easiest user interface first under the user interface title.

Extent and the Type of Use of AI

We would rank O-Plan at the top of list because it has the highest use of AI. Indeed, we found that users extensively use O-Plan for planning. NLA makes use of traditional, rule-based AI as a search engine. In our opinion, Web Watcher struggles in making an effective use of AI's machine-learning domain. From the experiment results and the constant changing of the algorithms, there are indications that the Web Watcher developers have not come up with a successful AI implementation.

Approach to Web-based AI

Fig. 1 : NLA

Fig. 2 : O-Plan

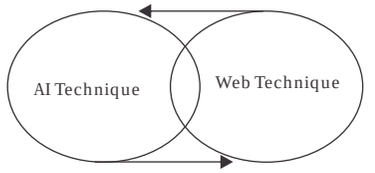


Fig. 3 : Web watcher

We illustrate the three agents described above based on how the agents use the AI technology and the Web technology. The three figures above show that NLA embeds AI into a web-based system, O-Plan embeds Web component into the AI system, while Web Watcher couples the Web and AI subsystems on a parallel interface.

Control

The greatest advantage of Web Watcher is that the user remains in control at all times. The user can decide to use or ignore the advice (suggested hyperlinks) provided by the agent. While using NLA-based AI agent, both the user and the system are in control as interacting partners, i.e. they both depend on the input from one another. This is unlike O-Plan, where once the user selects the input data, O-Plan takes control. Also, with O-Plan, most fields accept only the system-defined input. The user can only make a selection from the items provided in the drop-down menus. This scenario leaves the users with much less control over what they can specify.

Security

We were surprised to note that the extant literature did not address the security issue, especially with O-Plan which is supposed to be used by the US Army. In our opinion, security should be one of the greatest concerns with any web-based application. On the other hand, being a government-oriented project, the authors might have circumvented the issues of security so as to avoid exposing the vulnerabilities of the systems to future potential security bleaches.

Input type or user interface

Although we could not use Web Watcher, judging from the demo, it appeared to be an easy tool to use. We tried O-Plan and liked its user interface (http:// www.aiai.ed.ac.uk/ project/oplan/isd/password: show O-plan). However, some input sheets were not clear as to what was expected as the input; nevertheless, we liked the menu-driven part. Considering the variety of details it can process, O-Plan does a good job of providing user-friendly interface. NLA can understand both the natural language as well as the type-in input. Thus, NLA outperforms the other two agents in providing a wider variety of the accepted input type. This notwithstanding, from examining the article, it is clear that it takes some practice to get used to the way NLA works.

Product Maturity or Evolution

NLA had two predecessors, namely "Happy Assistant", that provided limited language processing, and its improved version after the first user study called "Natural Language Sales Assistant." Natural Language Sales Assistant is an incremental improvement of the Happy Assistant. The first work in this area seems to have been reported in the year 2001. Thus, this is a newer agent that has gone through the process of proof of

concept, prototyping, and the pilot deployment. Natural Language Sales Assistant was reported in summer 2002.

Web Watcher work was reported in 1995 and continued up to 1996, during which period the AI backbone was revised several times. However, the product's functionality seems to have remained the same since then and there is no evidence of any current work in this area. Based on what we came across, the initial ideas on O-Plan were published in1991. These ideas seem to have been used in a project (Enterprise project: 1993 to 1996) and then the joint project with DARPA and University of Edinburgh started in 1995 with automated planning support aid. The work is still going on; hence, of the three, this program has the longest successful life.

All the three agents did develop prototypes. We are not sure about the availability of NLA to the general public. Web Watcher is no longer supported on the Internet, but O-Plan provides in-depth access on the Internet.

Research experiments

The O-Plan has an edge over NLA and Web Watcher since the O-Plan provides more complete and modular set of tools. More specifically, the O-Plan can accommodate all three phases of completing a task including planning, assigning the command, and monitoring the task execution. The O-Plan can use different agents for these three roles, providing substantial modularity and flexibility. Using military logistics, the O-Plan project demonstrates how to use it with practical problems (O-Plan Approach, 2012). The O-Plan has already been applied to various projects including space-platform construction, satellite planning and control, non-combatant evacuation operations, and air campaign planning workflow. More data on these projects are available at http://www.aiai.ed.ac.uk/project/oplan/web-demo/index.html. Another advantage of O-Plan over other AI based planning systems is that the O-Plan adopts well with a dynamically changing environment by allowing refinement of an initial plan or requirements to regenerate new options (Tate& Dalton, 2003).

While O-Plan's strength lies in using it for practical problems, the strengths of NLA and Web Watcher lie in using them to perform experiments, resulting in statistical data. We were surprised not to see any such experiments in the existing literature on O-Plan. Only one of the papers (Tate, Dalton, & Levine,1998) from the eight articles (Tate, Levine, Jarvis, & Dalton, 2000; Tate, Dalton, & Levine 2000, Tate, Dalton, & Levine, 1998, Tate 1997a, Tate 1997b, Tate 1997c, Drabble & Tate, 1995; and Fraser & Tate 1995) lists a scenario describing an experiment to envisage system use. The papers fall short in providing the experimental data, such as the comparison of manual planning versus using the O-Plan.

Validation

All the AI agents were easily accessible for testing. O-Plan has been online at least since the year 2000 and is still actively supported. Web Watcher was also available on the Web. It provided 1,777 tours from August 1995 to March 1996. Through email correspondence between one of the authors and a caretaker person, we conclude that it seems like the project has come to a standstill now. NLA was deployed as pilot on IBM's website (time duration not mentioned).

CONCLUSION

Considering the growth potential of the World Wide Web, we looked at three AI agents that can be effectively used online in different domains such as commercial, military, or individual. We discussed the strengths of these agents and the issues that still need to be addressed. NLA can be viewed as a prelude to other, more advanced semantic-based natural language processing search engines that use AI. O-Plan is a promising agent that demonstrates a broad use of AI planning technologies across the whole planning cycle. The products, such as Web Watcher that implement machine learning, are promising agents for users to utilize while surfing on the Internet. The versatility and the value of deployment of AI technology over the Web could make AI a crucial part of the Web.

REFERENCES

1. *Armstrong, R., Freitag D., Joachims, T., and Mitchell, T. (1995) "Web Watcher: A Learning Apprentice for the World Wide Web." AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environment.*
2. *Chai, J.,Horvath, V., Nicolov, N., Stys, M., Kambhatla, N., Zadrozny,W.,and Melville, P.,(2002) "Natural Language Assistant: A Dialog System for Online Product Recommendation. "AI Mag. 23(2), 63–76.*
3. *Claburn, T., (2011) " Apple iPhone 4S: 5 Key Facts." InformationWeek, October 04, retrieved on 10/05/2011 from http://www.informationweek.com/news/mobility/smart_phones/231700257*
4. *CTIA, (2011), Wireless Quick Facts, Retrieved on 10/01/2011 from http://www.ctia.org/advocacy/research/index.cfm/AID/10323*
5. *Detlor B., and Arsenault C., (2002) "Web Information Seeking and Retrieval in Digital Library Contexts: Towards an Intelligent Agent Solution."Online Inf. Rev., 6:404–412.*
6. *Drabble, B., and Tate, A. (1995) "O-Plan: A Situated Planning Agent" Proceedings of the Third European Workshop on Planning (EWSP-95). Assisi, Italy, September.*
7. *Ernandes, M., Angelini, G., and Gori, M., (2008) "A Web-Based Agent Challenges Human Experts on Crosswords."AI Magazine, 29(1), 77-90.*
8. *Erwin, S. I., (2008) National Defense. Arlington: Vol. 92, Iss. 655; pg. 18.*
9. *Farber, D., (2008) Radar Networks Delivers Twine 1.0, Cnet News, October 20, 8:14 PM.*
10. *Fraser, J., and Tate, A., (1995) "The Enterprise Tool Set-An Open Enterprise Architecture."Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95). Montreal, Canada. August.*
11. *Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005) "Auto Tutor: An Intelligent Tutoring System with Mixed-initiative Dialogue." IEEE Transactions in Education, 48, 612–618.*
12. *Graesser, A. C., Jeon, M., and Dufty, D. (2008) "Agent Technologies Designed to Facilitate Interactive Knowledge Construction." Discourse Processes, 45, 298-322.*

13. Hinz, O., and Eckert, J. (2010) "The Impact of Search and Recommendation Systems on Sales in Electronic Commerce", Business & Information Systems Engineering, 2(2), 67-77.
14. Joachims, T., Freitag, D., and Mitchell, T. (1997) "Web Watcher: A Guide to the World Wide Web (shorter version)." Proceedings of IJCA97, August.
15. Joachims, T., Freitag, D., and Mitchell, T. (1996) "Web Watcher: A Guide to the World Wide Web (longer version)." CMU Technical Report, CMU-CS-96-X, September.
16. Joachims, T., Mitchell, T., Freitag, D., and Armstrong, R. (1995) "Web Watcher: Machine Learning and Hypertext, "Proceedings of IJCA97, August 1997. Dortmund, Germany, August.
17. Lopez, A., Jerome J., and Cleckner, W., (2004) "Machines, the Military, and Strategic Thought. "Military Review, September-October, pp. 71-77.
18. Martins, L. L., Gilson, L. L., and Maynard, M. T. (2004) "Virtual Teams: What Do We Know and Where Do We Go from Here?" Journal of Management (30:6), pp. 805-835.
19. Musliner, D.J., Durfee, E.H., and Shin, K.G. (1993) "CIRCA: A Cooperative Intelligent Real-time Control Architecture."IEEE Transactions on Systems, Man and Cybernetics, 23(6).
20. Musliner, D. J., Hendler, J. A., Agrawala, A. K., Durfee E. H., Strosnider J. K., and Paul C. J. (1995) "The Challenges of Real-Time AI."IEEE Computer, 28(1).
21. O-Plan Approach (2012), "Approach," Available at: http://www.aiai.ed.ac.uk/project/oplan/oplan/approach.html
22. Paul, I., (2011), IBM Watson Wins Jeopardy, Human Rally Back, PCWorld, Feb 17, 2011, retrieved on 10/05/2011 from http://www.pcworld.com/article/219900/ibm_watson_wins_jeopardy_humans_rally_back.html
23. Quinlan J.R. C4.5: (1993) Programs for Machine Learning. Morgan Kaufmann
24. Rissanen, J. (1978) "Modeling by Shortest Data Description."Automatica, 14
25. Sandra, I., Erwin[2] National Defense. Arlington: Jun 2008. 92(655), pp. 18.
26. Schoppers, M. (1994) "A Software Architecture for Hard Real-Time Execution of Automatically Synthesized Plans of Control Laws."Proceedings of AIAA/MASA conference on Intelligent Robots in Field, Factory, Service, and Space. March.
27. Shimazu, H. (2002) "Expert Clerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Web-shops," Artificial Intelligence Review 18, pp. 223-244.
28. Tate, A., Levine, J., Dalton, J. and Nixon, A. (2002) "Task Achieving Agents on the World Wide Web, In Creating the Semantic Web. "Fensel, D., Hendler, J., Liebermann, H. and Wahlster, W. (eds.), 2002, MIT Press.
29. Tate, A., Levine, J., Jarvis, P., and Dalton, J.,(2000a) "Using AI Planning Technology for Army Small Unit Operations." Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems (AIPS_2000), May.
30. Tate, A. and Dalton, J. (2003) "O-Plan: a Common Lisp Planning Web Service,"Proceedings of the International Lisp Conference, October 12-25, 2003, New York, NY.
31. Tate, A., Dalton, J., and Levine, J. (2000b) "O-Plan: a Web-based AI Planning Agent. "Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI-2000). Austin, Texas. August.
32. Tate, A., Dalton, J., Levine, J. (1998) "Generation of Multiple Qualitatively Different Plan Options." Proceedings of the Fourth International Conference on AI Planning Systems (AIPS '98). Menlo Park, California. 1998.
33. Joachims, T., Freitag, D., Mitchell, T. (1996) "Web Watcher: A Guide to the World Wide Web (longer version)."CMU Technical Report, CMU-CS-96-X, September.
34. Tate, A.,(1997a) "A Planning Agent on the World Wide Web." Based on AAAI-97 Workshop on Constraints and Agents. Providence, RI. July.
35. Tate, A. (1997b) "Mixed Initiative Interaction in O-Plan." Proceedings of AAAI Spring 1997 Symposium on Computational Models for Mixed Initiative Interaction. Stanford University, March.
36. Tate, A. (1997c) "Multi-agent Planning via Mutually Constraining the Space of Behavior ."Based on AAAI-97 Spring Symposium on Mixed Initiative Interaction, Stanford University, March.
37. Tate, A., (1995), "Characterizing Plans as a Set of Constraints - the <I-N-OVA> Model a Framework for Comparative Analysis, in the Special Issue on "Evaluation of Plans, Planners, and Planning Agents", ACM SIGART Bulletin Vol. No. 1.
38. US Census Bureau (2009), Computer and Internet Use, Retrieved from http://www.census.gov/hhes/computer/publications/2009.html on 10/02/2011
39. Wilson, D.C., Leland, S., Godwin, K., Baxter, A., Levy, A., Smart, J., Najjar, N. and Andaparambil, J. (2009) "Smart Choice: An Online Recommender System to Support Low-Income Families in Public School Choice."AI Magazine. 30(2), 46-59.