A Learning Apprentice For Browsing

Robert C. Holte **Chris Drummond**
Computer Science Department
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
{holte,cdrummon}@csi.uottawa.ca

Abstract

This paper describes the task of browsing and an agent we have developed to improve the speed and success rate of browsing. The agent is a learning apprentice: it monitors the user's normal browsing actions and learns a measure of "relevance" to the user interests. It searches the library being browsed, uses the learned measure to evaluate items and presents to the user those that are most relevant. The paper discusses the main issues raised during the development of the browsing agent. These are of general interest not only because browsing is of considerable practical importance but also because it represents a prototypical task for learning apprentice research.

1 THE BROWSING TASK

"Browsing" is the searching of a computer library for an individual library item. In the browsing task, the human doing the search (the "user") starts with a set of requirements, which could range from a complete formal specification to an informal list of desiderata. The aim of browsing is to find an item that best meets these requirements. Search proceeds by the user assessing the information about library items currently being displayed and choosing a "search" operation. Typical operations are various forms of content-based retrieval ("indexing"), requesting additional information about a particular item, and navigating along links in the library that connect related items. The operation is executed by the browsing system and the display updated. This process is repeated until the user deems an item to adequately satisfy the requirements or decides to abandon the search.

In the browsing task, the user is assumed to have incomplete, imperfect knowledge of the content, organization, and descriptive language of the library. Because of this, the browsing process is fundamentally uncertain and iterative. Although there might exist a single operation or short sequence of operations that will retrieve the "best" item, the user is, in general, uncertain of which operations are most useful and perhaps even uncertain of which item is "best". Browsing is therefore guided by the user's expectation of what the target item in the library will be like. We call the user's mental model of the target item the "search goal". Unlike the requirements, which remain fixed throughout a search, the search goal is continually being refined, perhaps considerably altered, as browsing proceeds and the user gains knowledge about the library.

2 AN AGENT TO ASSIST BROWSING

We have investigated browsing in the context of software reuse, in which the user browses a library of software looking for a particular module. The aim of our research is to increase the speed and success rate of browsing. Our approach has been to develop a browsing agent that can be "attached" to an ordinary browsing system to enhance the system's effectiveness. The user browses as usual, unaware that the agent is monitoring his actions. From the sequence of user actions the agent infers an "analogue" representing what it believes to be the user's search goal. The analogue is converted into a form that can be readily used to measure the relevance of an individual library item to the user. This relevancy measure is applied to each item in the library, and the items are sorted according to their relevance. The resulting information can be used in a variety of ways to influence the user: in the present implementation the sorted list is simply displayed to the user in a special window called the "Suggestion Box". The browsing agent is described in detail in [Drummond et al.,1993a].

To emphasize the characteristics of our agent that distinguish it from other systems for improving browsing, we describe it as an active, autonomous assistant.

Active: the agent is continuously analyzing the user's actions, searching the library, and attempting to improve the speed and success rate of browsing. By contrast, most other systems are passive, in the sense that they operate only when explicitly invoked by the user for a particular purpose.

Autonomous: the agent's activities are completely transparent to the user. The user browses in the normal manner: no special input is directed towards the agent, and the feedback from the agent is fully integrated with the normal display of the browsing system.

Assistant: the agent is not an essential part of the browsing system; without the agent the browsing system remains fully functional.

The browsing agent achieves its autonomy by learning how to measure "relevance" from the user's normal browsing actions. The term "learning apprentice" [Mitchell et al.,1985] is used for active, autonomous assistants that learn. Several learning apprentices have recently been developed for various tasks: scheduling meetings [Dent et al.,1992; Maes and Kozierok,1993], filling-in forms [Hermens and Schlimmer,1993], and note-taking [Schlimmer and Hermens,1993].

Experiments with our browsing agent have been run with simulated users (see "Experimental Methodology" below). The goal of the first experiment was to measure the accuracy of the inferred relevancy measure; in this experiment the simulated user was not given access to the suggestion box. About 40% of the time the library item for which the user was searching (the "target") was inferred by the browsing agent before it was found by the user. Furthermore, during the course of the search, the agent consistently ranked the target significantly higher than the user. A second experiment measured the effect of the suggestion box on browsing time. The occasional use of the suggestion box reduced browsing time for 52 of the 189 targets and increased it for about 40 of the targets. Details of the experiments are given in [Drummond et al.,1993a] and [Drummond et al.,1993b].

3 BROWSING AS A TESTBED TASK

[Sheth and Maes,1993] use "information filtering" as the testbed task for their research on learning agents. This is an excellent choice as a testbed for several reasons.

Research issues and methodology: In information filtering most of the major issues in learning apprentice research arise, but can be isolated and varied through the full range of difficulty for experimental purposes.

Feasibility: Initial studies, e.g. [Sheth and Maes,1993] in the domain of USENET news articles and our browsing experiments described above, show that the task is not impossibly difficult: although the task raises many challenging research issues, successful applications can be expected in the relatively near future.

Practical importance: Information filtering is a serious practical problem with many large-payoff applications. Its significance is immediately appreciated by all members of the computing industry.

Browsing is a constrained form of information filtering that shares all of these attractive features. Browsing is more constrained than general information filtering because the information to be filtered, i.e. the library contents, changes slowly and tends to be reasonably well structured, and, more importantly, because users tend to be much more narrowly focused while browsing than, say, while reading news. The browsing task therefore eliminates some of the complexities of the general task while retaining all its advantages.

The remaining sections of the paper introduce the main issues in learning apprentice research and discuss the form they take in the browsing task and how each issue is addressed in our current research.

4 HUMAN-COMPUTER INTERFACE

Learning apprentices almost invariably are used to speedup the problem-solving of a human involved in "interactive search". Interactive search is human-computer co-operative problem-solving in which the human is the controlling partner, deciding which actions to execute and when the goal has been reached. Interactive search is natural for any task in which the search space itself is well-defined, but the search goals are ill-defined. Information filtering and browsing are interactive search tasks.

The fact that a human, not a computer program, is controlling the search is the source of several challenging issues for learning apprentice research. The most obvious is that, because the user decides which actions to take, a learning apprentice system cannot dictate the search strategy. It can only try to influence the user's decisions by altering the form and content of the information displayed to the user. The challenge is to be "convenient yet unobtrusive" (p.63, [Schlimmer and Hermens,1993]) – to exert sufficient influence to change the user's search behaviour while avoiding excessive influence that distracts or impedes the user in carrying out the task.

Most learning apprentices to date have influenced the user by presenting suggestions. In the form-filling task [Hermens and Schlimmer,1993] there is virtually no penalty incurred by wrong suggestions. But in browsing, as in most tasks, the user must pause to consider suggestions presented to him, and therefore a significant cost in terms of time and distraction is incurred. Our approach has been to keep suggestions out of sight but easily accessible so that they are easily ignored when the user is progressing satisfactorily and easily consulted when the need arises.

5 DEGREE OF AUTONOMY

We have defined a learning apprentice as being completely autonomous, learning only from the user's normal behaviour. In general, the user's actions do not explicitly, unambiguously convey the goal ("classification") information that is needed for learning, nor do they indicate the factors justifying (in the user's view) the choice of a particular action. The fact that goal information is imperfect means that it is necessary to infer the user's goals; the fact that justifications are not given makes accurate goal inference difficult.

One approach to this problem is to relax the requirement for full autonomy by permitting the learning apprentice to ask questions about the classification of an item ("is this interesting ?"), as in [Sheth and Maes,1993], perhaps accompanied by explanations, as in Protos [Porter et al.,1990], or to request a critique of an item ("relevance feedback" [Harman,1992]). It is important to realize that this additional input only reduces the problem, it does not entirely solve it. The explicit classifications and justifications provided by the user are, at best, only fragmentary and approximate statements of the true goals and justifications, and may become invalid if the goals shift in the course of browsing. Consequently, it is not clear whether the reduction in autonomy is cost-effective: the additional information acquired may not compensate the time penalty incurred by the acquisition process.

Our approach to this problem is to maintain full autonomy by developing methods of goal inference and tracking.

6 FEATURE ENGINEERING

Although the user's actions do not explicitly, unambiguously convey information about the user's goal or justifications for actions, there is no doubt that they IMPLICITLY carry such information: this is a consequence of the fact that the actions have been deliberately chosen, to the best of the user's ability, to progress towards the goal. However, the number of actions required in order to infer the goal will heavily depend on HOW explicitly and unambiguously the actions convey information about the goal. This is analogous to the fact, well-known in machine learning, that the speed of learning depends on the quality of the features with which examples are described. Like a "good" feature, a "good" action is informative, or unambiguous, in the sense that there are relatively few reasons for a user to execute it.

Based on this insight, we analyzed the actions available in the original browsing system. Many were highly ambiguous. These we replaced with less ambiguous actions that maintained or enhanced the system's functionality. In re-engineering the browsing system, we were careful to consider the effects of a change on browsing as well as on goal inference; we avoided any change that would degrade the normal browsing effectiveness, no matter how much the change would improve goal inference. Most actions in the re-engineered system are highly informative, and the justifications for executing each action are a key part of the knowledge base used by the browsing agent to infer the search goal analogue from the user's actions.

Conceptually, a justification for an action is of the form "action X was selected because the user is interested in items with feature Y (which action X emphasizes)". Each action produces one or more justifications, which are used as training examples for learning. The usefulness of these training examples ultimately depends upon the quality of the basic "features" of the library items, i.e. the information stored in the item (such as its name) and the links to other items. Some of these features – keywords, for example, and structuring information ("IS-A") – are explicitly intended to assist browsing, but it is invariably the case that the user's true goal and many of the factors that affect his search control decisions will not be representable/computable on the basis of the available features. [Dent et al.,1992] makes a similar observation.

We have done no engineering of the basic library features, believing that most libraries are too large and amorphous, and used by too diverse a community of users for it to be feasible to engineer an adequate set of features. However, we have included in the browsing agent knowledge for inferring additional features of an item based on the item's own features and the features of items related to it in specific ways.

7 CONTEXT-DEPENDENT LEARNING

The purpose of a learning apprentice is sometimes described as "personalizing" [Sheth and Maes,1993] or "customizing" [Schlimmer and Hermens,1993] a generic software system. From this description it is clear that what is learned is meant to be of limited scope: it is learned for one specific user and is not transferrable to any other.

The "temporal scope" of learned knowledge may also be limited. The browsing agent described above learns about the user's current search goal; as soon as the search ends, all the learned knowledge is obsolete. To succeed at this task, the learning apprentice must truly learn in real-time ([Schlimmer and Hermens,1993] erroneously uses "real-time" when it means "on-line"). This is an extreme type of temporal scope, but most learning apprentices will face a similar (usually longer term) time constraint because in most tasks users' interests change with time. Only recently has research begun into learning concepts that change with time (or with other contextual factors) [Turney,1993a,1993b; Kilander and Jansson,1993; Widmer and Kubat, 1993]. In some domains concepts (or user's interests) may even be periodic, shifting cyclically between several alternatives [Widmer and Kubat, 1993; Dent et al.,1992].

In many domains, each user has multiple simultaneous interests/goals, among which there is no transfer of learned information. For example, in the note-taking domain [Schlimmer and hermens,1993] a user may write notes on any number of distinct subjects. In order for the agent to learn which word the user will write next it must know which subject is being written about.

8 ACCURACY VS. COVERAGE

Because a learning apprentice is an autonomous assistant, there is no penalty incurred if it does nothing – the user proceeds normally, without even being aware of any (possible) lost opportunities for speedup. This observation underlies the strategy for improving the accuracy of a learning apprentice by having it act only when it is relatively confident it is correct. This is called trading coverage for accuracy: the agent makes fewer predictions (lower coverage) but the predictions it makes are more accurate. [Schlimmer and Hermens,1993] employ a variant of this strategy in which the agent informs the user of its confidence.

[Holte et al.,1989] found that a moderate sacrifice in coverage could produce a relatively large gain in accuracy: in the three sets of results reported there (Tables 2 and 3), error rate could be cut in half by reducing coverage 12%, 19% and 38%. Similar findings are reported in [Danyluk and Provost,1993], where error rate could be cut in half by reducing coverage by 18% (Table 2) and 24% (Table 4). In [Dent et al.,1992] error rate could be cut in half by reducing coverage about 40%.

9 MULTI-AGENT INTERACTION

If agents are fully autonomous, there are only two types of interaction that can occur between multiple agents in the same environment. The first type of interaction is the simplest, raising no new issues. If one agent calls upon a system to perform part of its processing, a second agent could be attached to that system; neither agent would be aware of the other's existence. For example, the user of the browsing system was a human in the scenario described earlier, but it could just as well have been another agent.

The other type of interaction arises when two or more agents are attached to the same system and give "advice" of the same kind. The difficulty is that the advice produced by one agent may conflict with, or in other ways differ from, the advice produced by another. One could simply have each agent present its advice separately, leaving it to the user to detect and reconcile differences. but this could be so large a burden on the user that the potential benefits of multiple agents is lost. The alternative is to have a co-ordination system that combines the advice of all the related agents into a coherent set of recommendations to be presented to the user.

Our research to date has involved only a single agent, but there are additional agents for the browsing task that could be developed and run in parallel with ours. For example, [Henninger,1991] describes an agent that assists the user in reformulating a query. As a second example, the response time of a browsing system could be improved by an agent that infers which actions the user is most likely to execute next, and precomputes the results of these actions. Another agent that would be useful is one that determines the best way to influence a particular user.

10 GENERALITY

The issue of generality is this: to what extent can an agent be independent of the specific environment in which it operates ? For example, the current implementation of our browsing agent was written for a specific browsing system and a specific library of object-oriented software. Although some of its knowledge is necessarily system-specific (e.g. the rules derived from the justifications of each action supported by the system), and the code implementing the agent is intertwined with the browsing system's code, it is our hope that much of the knowledge and basic design of the agent is system- and library-independent, and applicable to browsing generally and perhaps to the more general tasks of information filtering or interactive search. Our present research is an initial investigation of the generality of our agent.

The most vexing question concerns the independence of the agent from a particular system. How can the presentation of the agent's "advice" to the user be fully integrated with the normal display of the system (part of the definition of "autonomy") and yet system-independent ? When we speak of "attaching" an agent to a system, it sounds like a very simple operation. But the fact is that coupling an agent and system involves knowledge engineering (e.g. about the actions and about features/links in the library), designing a "natural" and effective means of presenting the agent's advice, and perhaps re-engineering of the system's actions/human-interface.

11 EXPERIMENTAL METHODOLOGY

In the normal experimental methodology, the speed of a problem-solver without a learning component is compared to the speed on the same set of problems of the same problem-solver with a learning component added. Depending on the goals of the experiment, the benchmark problems may be drawn from a "real-world" domain or generated artificially.

This methodology can be awkward to apply when the problem-solver is a human. The first difficulty is that every human is different, and experimentation must ultimately show that speedup is obtained over a broad range of problem-solvers (i.e. humans), not just one. Secondly, humans learn from their problem-solving experience, so one does not get a valid comparison by "running" the human on the same problems with and without the aid of a learning component. Thirdly, the fact that the learning system cannot directly control search, but must influence search through a human user, means that the system may correctly infer ways to speedup search, yet fail to achieve any speedup because it did not convince the human to act on its advice. Experimentation must distinguish this failure mode from the alternative, classical failure mode, in which the system fails to infer a way to speedup search. Finally, the use of human subjects gives rise to the same experimental difficulties that arise when using real data in ordinary machine learning experimentation: absence of experimental control and, often, limited availability (in terms of number or time) of subjects.

These difficulties are not insurmountable in some domains. Good "real world" experiments have been done in the form-filling domain [Hermens and Schlimmer,1993] and in the meeting-scheduling domain [Dent et al.,1992]. The information filtering task, including browsing, is well-suited to "real world" experiments. A wide variety of "real world" libraries and browsing systems are readily available, making it easy to experiment with a diverse range of problems in a true applications setting. Often there is also available a large pool of subjects (people already familiar with and regularly using the library and browsing system of interest).

An alternative experimental methodology is to use "simulated" users, possibly in an artificial setting. This overcomes all of the above difficulties, but raises the difficulty of creating simulated users sufficiently realistic that the results of the experiments are relevant to the real application setting. For most, perhaps all tasks, browsing included, it is infeasible to create simulated users that exhibit the rich behavioural patterns of a human user. Our aim in creating simulated users for browsing has been to try and simulate some of the most prominent general behavioural trends that a human might be expected to follow.

Our simulated users consist of two parts: a "fuzzy oracle" that represents the search goal, and a heuristic search strategy that consults the oracle and selects browsing actions. The heuristic search strategy is a combination of depth-first search and hill-climbing. The fuzzy oracle contains a target class selected by the experimenter from amongst the classes in the library. The oracle gives YES/NO answers to questions about whether a given library item matches the target class in certain ways. The oracle is "fuzzy" because its answers are not always correct; for each type of question, the experimenter can set the probability that the oracle will give an incorrect response. This noisiness represents the user's uncertainty in evaluating the degree of match between a library item and his requirements. The noisiness is reduced as search progresses to simulate the user's growing certainty about his goal.

12 SUMMARY

This paper has described the task of browsing and an agent we have developed to improve the speed and success rate of browsing. The agent is a learning apprentice: it monitors the user's normal browsing actions and learns a measure of "relevance" that can be applied to items in the library being browsed. Using this measure, it searches the library and presents to the user the items is finds to be most relevant. The paper has briefly discussed the issues that arose during the design of the browsing agent; these issues include all the main issues in learning apprentice research. From this, and the fact that browsing is a feasible task of considerable practical importance, we conclude that browsing is a good testbed for learning apprentice research.

Acknowledgements

This research was supported in part by a strategic grant ("Machine Learning Applied to Software Reuse", held jointly with S. Matwin and F. Oppacher) and an operating grant from the Natural Sciences and Engineering Research Council of Canada. We wish to thank D. Ionescu (Electrical Engineering Dept., University of Ottawa) for his support of the active browsing project.

References

Danyluk, A.P. and F.J. Provost (1993). Small Disjuncts in Action: Learning to Diagnose Errors in the Local Loop of the Telephone Network. *Proceedings of the 10th International Conference on Machine Learning*, P. Utgoff (editor), pp. 81-88.

Dent, L., J. Boticario, J. McDermott, T.M. Mitchell and D. Zabowski (1992). A Personal Learning Apprentice. *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI'92)*, pp. 96-102.

Drummond, C., D. Ionescu and R.C. Holte (1993a). Accelerating the Searching of Software Libraries through Goal Inference, technical report TR-93-19, Computer Science Dept., University of Ottawa.

Drummond, C., R.C. Holte and D. Ionescu (1993b). Accelerating Browsing by Automatically Inferring a User's Search Goal, *Proceedings of the Conference on Knowledge-Based Software Engineering*, pp. 160-167.

Harman, D. (1992). Relevance Feedback Revisited. *Proc. 15th International Conference on Research and Development in Information Retrieval (SIGIR'92)*, pp. 1-10.

Henninger, S. (1991). CodeFinder: A Tool For Locating Software Objects For Reuse. *Automating Software Design: Interactive Design Workshop of the 9th National Conference on Artificial Intelligence (AAAI'91)*, pp. 40-47.

Hermens, L.A., and J.C. Schlimmer (1993). A machine-learning apprentice for the completion of repetitive forms. *Proc. 9th Conference on Artificial Intelligence for Applications*, pp. 164-170.

Holte, R.C., L. Acker, and B.W. Porter (1989). Concept Learning and the Problem of Small Disjuncts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 813–818. Morgan Kaufmann.

Kilander, F. and C.G. Jansson (1993). COBBIT - A Control Procedure for COBWEB in the Presence of Concept Drift. *Proceedings of the European Conference on Machine Learning (ECML-93)*, Pavel Brazdil (editor), pp. 244-261.

Maes, P., and R. Kozierok (1993). Learning Interface Agents, *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI'93)*, pp. 459-465.

Mitchell, T.M., S. Mahadevan and L. Steinberg (1985). LEAP: A Learning Apprentice for VLSI Design. *IJCAI'85*, pp. 573-580.

Porter, B.W., E.R. Bareiss, and R.C. Holte (1990). Concept Learning and Heuristic Classification in Weak-Theory Domains. *Artificial Intelligence*, 45(1-2):229-263.

Schlimmer, J.C. and L.A. Hermens, (1993). Software Agents: Completing Patterns and Constructing User Interfaces. *Journal of Artificial Intelligence Research*, vol. 1, pp. 61-89.

Sheth, B. and P. Maes (1993). Evolving Agents For Personalized Information Filtering. *Proceeding of the 9th Conference on Artificial Intelligence For Applications*, pp. 345-352.

Turney, P.D. (1993a). Exploiting Context when Learning to Classify. *Proceedings of the European Conference on Machine Learning (ECML-93)*, Pavel Brazdil (editor), pp. 402-407.

Turney, P.D. (1993b). Robust Classification with Context-Sensitive Features. *Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-93)*, P.W.H. Chung, G. Lovegrove, and M. Ali (editors), Gordon and Breach Science Publishers, Switzerland, pp. 268-276.

Widmer, G. and M. Kubat (1993). Effective Learning in Dynamic Environments by Explicit Context Tracking. *Proceedings of the European Conference on Machine Learning (ECML-93)*, Pavel Brazdil (editor), pp. 227-243.