Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Providing Web Surfers With Useful Information About Plants via PlantSage

W. Potter
University of Georgia

E. Shamblen
University of Georgia

H. Vuppula
University of Georgia

Providing Web Surfers With Useful Information About Plants via PlantSage

W. D. Potter, E. Shamblen, and H. Vuppula
Artificial Intelligence Center
GSRC 111, Brooks Dr.
University of Georgia
potter@cs.uga.edu

Abstract

The World Wide Web has opened substantial avenues for research into accessing and disseminating vast amounts of information to millions of users all over the world. As a result, it is now very common for people at all levels of computer literacy to “surf the net” in search of information. However, finding useful information can be a laborious and tedious task, especially for those unfamiliar with the Web. Our goal is to improve both the usefulness and the ease of web-based information gathering. In particular, we place a special emphasis on horticulture-based information. In the PlantSage project, we provide access to three different sources of information: a local database, a search on selected “partner” sites, and a search on the entire web that not only returns names and URLs of relevant sites but also collates and uniformly formats potentially useful information. This paper will describe each of these processes and our plans for enhancing them.

I. Introduction

We started this project with the following goal in mind: provide useful information about plants. We began by taking a three-pronged approach to obtaining plant information. Our approach begins with a web surfer specifying a plant name (either scientific or common from either an index of names or by typing in a name). The first source of information is a local database of plants containing certain growth characteristic data and some photographs. We provide access to over 15,000 plants in our local database. Our second source of information comes from pre-determined (partner) web sites that provide specific detailed plant information, such as the USDA Integrated Taxonomic Information System (ITIS) (http://www.itis.usda.gov/plantproj/itis/index.html) or the Missouri Botanical Garden database (http://www.mobot.org/). Third, the rest of the web provides additional information. But what makes all this useful?

Clearly, the "local" and "partner" information sources provide useful information that can easily be tailored to various types of web surfer (from researcher to commercial user to casual visitor). The remaining "web" source provides the greatest challenge to obtaining useful information. To attempt this challenge, we pursued the development of an intelligent web search agent [Bigu97, Brad91, Knap98]. Our hopes were to implement an intelligent agent that: (1) finds web sites containing information about a plant, (2) analyzes the available information, (3) extracts the relevant portions, and (4) synthesizes these portions into a presentable collection of plant knowledge.

II. Initial Design

From the very beginning of the project, we envisioned a multi-dimensional information provider approach with orthogonal characteristics. Namely, we felt that web surfers could be generally categorized into three types: the "casual visitor" which includes home gardeners and causal surfers, the "educational visitor" which includes students, teachers, and researchers interested in horticulture, and the "professional visitor" which includes landscape designers and nursery personnel. At right angles to this categorization is the level of information obtainable that would provide the proper amount of information breadth and depth. The categorization approach is based on characteristics the surfer sets when visiting the site and on characteristics determined automatically by the site when surfers visit.

Progress on the breadth and depth aspects is more visible and led to the notion of local, partner, and global (or web) views of information. For local information we felt that a surfer would want quick summary information about specific plants and to have the ability to view one or more images of the plants. We followed the example set by PlantLink (http://www.plantamerica.com) where surfers provide the name of a specific plant by either typing in the scientific name or selecting the scientific name from an extensive list. The ability to access scientific names using a search based on common names is available also.

If more information is desired than is provided by the local search, a surfer may request more depth on a specific plant by selecting the partner search option. Pre-selected web sites with plant information were identified as the partner sites. Our search system builds a partner query based on the specific plant and the partner search strategy (that is, the syntax of the partner search engine) and submits the query to the partner site. The main advantage to this approach is that the surfer may get specific information from a partner site without having to surf through a possibly large number of web pages to find the information. In essence, we have already done that and have given our surfer a short cut to the partner information.

A surfer still may not be satisfied with the amount of information presented by the local and partner searches. In this case, the surfer has the ability to automatically request information from the entire world wide web. We did not try to re-invent the search engine but instead use a standard engine that is already available. However, we feel that search engines are very rudimentary since they typically only provide links to web sites. We extend this functionality by extracting what we call useful information from the web sites identified by the search engine. That is, when the surfer selects the web search option, the plant name is sent as the search key to a search engine. The search engine finds web sites containing information about the plant. The sites are returned to our system, where our system inspects the information available from the sites and extracts useful text. The results are returned to our surfer along with the site links as references. If the surfer wants to visit one of the sites, they can simply follow the link.

III. Local Data and Images

Plant entries in our local database came from Michael A. Dirr's Photo-Library of Woody Landscape Plants on CD-ROM, Allan M. Armitage's Photo-Library of Herbaceous Plants on CD-ROM, and The Interactive Guide to Herbaceous Perennial Plants by Allan M. Armitage. The Interactive Guide contains numerous growth characteristic data on perennial plants and some images. The images that can be viewed on the PlantSage web site (http://dendrite.ai.uga.edu/plant/index.html) come from Allan M. Armitage's Photo-Library of Herbaceous Plants on CD-ROM. We extracted the data from the CDs and loaded it into a relational database on our server. The local processing uses java and JDBC (java database connectivity) to access the data during a surfer query [Naug97, Rees97].

IV. The Partner Search

Partners are those web sites that contain their own plant information and offer a provision for searching that information via the Internet. Since there are many sites that provide such information about plants, the question of how to decide which ones to include as partners was raised. It was decided to select only those sites that we have found to provide extensive and varied types of information, a group that will continue to grow as more such sites appear on the web.

We have a dedicated servlet (server side application program) for each of the partner sites to query their database [Moss98]. These servlets operate from our server and automatically access the partner web sites. Normally all of these web sites use CGI scripts to query their database. Our job is to figure out the URLs of these query programs and the attributes for querying them. For this, the source code of the query page has to be checked. Once we understand the URL, our servlet establishes the connection with that location, using the URLConnection class within java [Naug97]. The abstract class URLConnection is the superclass of all classes that represent a communications link between an application and a URL. Instances of this class can be used both to read from and to write to the resource referenced by the URL. Results pages of different partners are usually in different formats. We provide a uniform look for our page when we display the results. All of the servlets in the PlantSage web site use a standard page servlet class to generate page headers and page footers so that every page looks similar. The output from our query to a partner's site is tailored and only the required information is displayed.

V. The WebSearch and URLParser

WebSearch operates in a fashion similar to the Partner Search servlet, except that instead of using the URLConnection class to send a query to sites already known to have both search capabilities and plant information, it queries a search engine, in this case InferenceFind (http://www.infind.com). InferenceFind then returns the URLs of all sites that contain some mention of the scientific name being queried, as well as some extraneous general information sites. WebSearch eliminates some of these extraneous sites by using knowledge about how InferenceFind organizes its results, but some may filter through this first line of extraction.

Once the URLs have been collected, URLParser establishes a connection to each in turn, first checking to ensure the address is valid and then checking the source code of the URL to ensure that the plant name is indeed contained within it. This typically eliminates the extraneous sites that WebSearch let through. Next come the analysis and information extraction stages, which must deal with a difficulty not encountered in a partner search: namely, we do not know what format any of these sites use to present their information. This means that any heuristic we design to analyze and extract information would be likely to either miss some useful information or collect information that is less than useful. It was decided to let URLParser exclude some information and let the surfer choose to visit the site directly for anything that was missed rather than overwhelm the surfer with too much detail.

URLParser divides the source code of a URL into chunks of plain text, eliminating any hypertext, graphics and paragraph breaks in the process. All of the non-text items were considered unnecessary and would interfere with any attempt at a uniform presentation. The size of each chunk is determined simply by the amount of text between those things that were eliminated. The chunks are checked for the plant name. If the plant name is found, the chunk to be added to an ordered list. When a site has been fully analyzed, URLParser returns the list to the WebSearch servlet which presents it along with the site link to our surfer. This process is repeated until all URLs gathered by WebSearch have been parsed. In this way, the surfer is given a fairly uniform presentation of useful information and the ability to directly visit the sites from which that information was extracted.

VII. PlantSage: Phase 2.

In the partner search and web search directions, there is still much to be done. Partner sites are appearing on the web very rapidly, as are other web sites dealing with plant information. The notion of an over-arching or meta-browser facility to access this information and provide it to users in a useful fashion is still valid. However, as we have seen from this project, it is extremely difficult. One possible addition to the toolkit we have used to tackle this situation could be the development of a plant ontology meta-search intelligent agent. This could help with the problem of additional web sites becoming available; however, there is still the lingering problem of a generic site parser. That is, the problem of extracting useful information from a web site with an unknown format is yielding slowly to our approach. We are continuing to pursue knowledge-based approaches to parsing new web site information.

Acknowledgement

This project was funded by a grant from the Webel Fund. Our sincere thanks to Mr. Richard Webel of Innocenti & Webel, and to Mr. Roger Milliken of Milliken & Company for their encouragement, guidance, and support.

References

[Bigu97] Bigus, J.P., and J. Bigus, *Constructing Intelligent Agents with Java*, Wiley Publishing, 1997.

[Brad91] Bradshaw, J.M., *Software Agents*, AAAI Press, 1991.

[Knap98] Knapik, M., and J. Johnson, *Developing Intelligent Agents for Distributed Systems*, McGraw Hill Publishing, New York, 1998.

[Moss98] Moss, K., *JAVA Servlets*, McGraw Hill Publishing, New York, 1998.

[Naug97] Naughton, P., and H. Schildt, *JAVA: The Complete Reference*, 1997.

[Rees97] Reese, G., *Database Programming with JDBC and JAVA*, O'Rielly Series, 1997.