Comparison Shopping Agent Systems- ShopBot

2000. 5. 17.
99419-514
박상록

Question

● Explain the learning phase of ShopBot.

Contents

● A scalable comparison-shopping agent for the World-Wide Web
  Autonomous Agents 97, Robert B. Doorenbos, Oren Etzioni, and Daniel S. Weld
  • Intelligent Web agents
  • Online shopping agents
  • ShopBot

● A more scalable comparison shopping
  agent Engineering of Intelligent Systems(EIS 2000),
  HCI2000 J. Yang, J. Choi, J. Kim, H. Ham, K. Lee

Intelligent Web agents

● Design issue
  • Ability: To what extent can intelligent agents understand information published at Web site?
  • Utility: Is the agent’s understanding sufficient to provided genuinely useful assistance to users?
  • Scalability: Can the agent automatically extract information from unfamiliar Web sites?
  • Environmental Constraint: What properties of Web sites underlie the agent’s competence?

Online shopping agents

● Requirements of shopping agents
  • Help the user decide what product to buy.
  • Find specifications and reviews of the products.
  • Make recommendations.
  • Comparison shopping to find the best price for the desired product.
  • Monitor “What’s new” lists and other sources to discover new relevant online information sources.
  • Watch for special offers and discounts.

ShopBot(1)

● About ShopBot
  • Fully-implemented, domain-independent comparison-shopping agent.
  • Given the home pages of several online stores, autonomously learn how to shop at those vendors.
  • After learning, it is able to speedily visit shops, extract product information, and summarize the results for the users.
  • Require only minimal knowledge about different product domains without natural language processing.
  • Heuristic search, pattern matching, inductive learning techniques.
ShopBot(2)

- Operate in 2 phases
  - Learning phase
    - Offline learner creates a vendor description for each merchant.
  - Comparison-shopping phase
    - Real-time shopper uses these descriptions to help a person decide which store offers the best price for a given product.

ShopBot(3)

- Learning phase
  - Analyze online vendor sites to learn a symbolic description of each site.
  - Domain model
    - Example products
      - \( P_1, P_2, \ldots, P_n \)
    - Attributes of the products
      - \( \text{manufacture}(P) = \text{MS} \)
      - \( \text{name}(P) = \text{Win95} \)

ShopBot(4)

- Comparison-shopping phase
  - Use learned vendor descriptions to shop at each site and find the best price for a specific product.
  - Execute the extraction procedures found by the learner for a variety of vendors.

ShopBot(5)

- Environmental Regularities
  - The navigation regularity
    - Online stores are designed so consumers can find things quickly. Ex: searchable index, form.
  - The uniformity regularity
    - Use a uniform look and feel.
  - The vertical separation regularity
  - Online vendors obey these regularities because they facilitate sale to human users.

Create vendor description(1)

- Vendor description
  - The URL of a page containing a form for a searchable index.
  - A function mapping product attributes to fields of that form.
  - Functions for extracting product data from pages returned by the index:
    - Recognize failure pages
    - Strip header and tail information from successful pages.
    - Extract a set of individual product descriptions

Create vendor description(2)

- Problem of extracting product description
  - Web page typically contains not only one or more product descriptions, but also information about the store itself, meta-information about shopping process, headings, links to related sites, and advertisements.
  - Meta-information
    - “Your search for Windows matched 3 items”
    - “Your shopping basket is empty”
  - Unsupervised learning
Problems of learning a vendor description
- Identifying an appropriate search form
- Determining how to fill in the form
- Discerning the format of product descriptions in pages returned from the form
- Interdependent
  - The learner cannot be sure that a certain search form is the appropriate one until it knows it can fill it in and understand the resulting pages.

Overview
- Find a set of candidate forms.
- For each form $F_i$, compute an estimate $E_i$ of how successful the comparison-shopping phase would be.
- Determine how to fill in the form, then make several test queries using the form to search for several popular products.
  - Training examples for the product descriptions
  - Compute $E_i$

Finding and analyzing candidate forms
- Start at the vendor’s home page and follows URL links, performing a heuristic search looking for any HTML forms.
- Discard forms that are clearly not searchable indices. (user name, address, phone number)
- Associate matching attribute with the field.
  - Domain description contains synonyms for each attribute.
  - It matches the text preceding a field.

To determine the format of product descriptions in pages returned from the form
3 subproblems
- Learn a generalized failure template
  - Queries for dummy products
  - Build a generalized failure template
- Learn to strip out irrelevant header and tail information
- Learn product description formats

Success page
- Queries for several popular products
- Test result page for failure template
- If majority of result page is failure, the form is not appropriate search form for the vendor
- Otherwise, learner records generalized templates for the header and tail of success pages

Extracting product description
- Assume all product descriptions have the same format at a certain level of abstraction (uniformity regularity)
- Abstract language: HTML tag + keyword text
- Every product description starts on a fresh line.
  - Break body of each result page into logical lines
- Heuristic ranking function
  - sum of the number of lines, the number of price was found, the number of required attributes found
Generation the Vendor Description

1. Decide which form is the best one
   - Repeat previous process for every form.
   - Based on making an estimate $E_i$
   - $E_i$ reflects both the number of the popular products that were found and the amount of information present about each one.
   - Once learner has chosen a form, it records a vendor description for future.

Real-Time Comparison Shopping

1. The operation of the shopper
   - Once it has received a request from the user via the GUI, it goes in parallel to each online vendor’s searchable index, and fills out and submits the forms.
   - Find successful page, and strip off the header and tail, find production description, sorts the results by ascending order of price, and generate a summary for the user.

Empirical Results

1. Evaluate ShopBot Utility
   - Experiments
     - 3 groups
       - Those who used ShopBot (3)
       - Those who used Netscape’s search tools and were also given the URLs of 12 software stores used by ShopBot(2)
       - Those who were limited to Netscape’s search tools(2)

<table>
<thead>
<tr>
<th>Group</th>
<th>Time (Sec)</th>
<th>Purchase</th>
<th>Actual</th>
<th>Error</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.00</td>
<td>93.75</td>
<td>87.75</td>
<td>6.00</td>
<td>42.05</td>
</tr>
<tr>
<td>2</td>
<td>112.50</td>
<td>89.00</td>
<td>92.00</td>
<td>3.00</td>
<td>42.05</td>
</tr>
<tr>
<td>3</td>
<td>50.00</td>
<td>69.00</td>
<td>60.00</td>
<td>9.00</td>
<td>42.05</td>
</tr>
</tbody>
</table>

Table 3: Subjects using the ShopBot performed the task at cost and generally found lower prices.

Empirical Results(2)

1. Acquisition of new software vendors

<table>
<thead>
<tr>
<th>Vendor</th>
<th>URL</th>
<th>Price</th>
<th>Offered</th>
<th>Profit</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acme</td>
<td><a href="http://www.acme.com/">http://www.acme.com/</a></td>
<td>$50.00</td>
<td>40.00</td>
<td>10.00</td>
<td>40.00</td>
</tr>
<tr>
<td>Beta</td>
<td><a href="http://www.beta.com/">http://www.beta.com/</a></td>
<td>$75.00</td>
<td>60.00</td>
<td>15.00</td>
<td>45.00</td>
</tr>
<tr>
<td>Gamma</td>
<td><a href="http://www.gamma.com/">http://www.gamma.com/</a></td>
<td>$100.00</td>
<td>90.00</td>
<td>10.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Delta</td>
<td><a href="http://www.delta.com/">http://www.delta.com/</a></td>
<td>$125.00</td>
<td>100.00</td>
<td>25.00</td>
<td>60.00</td>
</tr>
<tr>
<td>Echo</td>
<td><a href="http://www.echo.com/">http://www.echo.com/</a></td>
<td>$150.00</td>
<td>110.00</td>
<td>40.00</td>
<td>60.00</td>
</tr>
</tbody>
</table>

Table 4: Prices found for DeltaSoft for the first and products at twelve software vendors. A price bold indicates that ShopBot correctly identified that the vendor was not selling that product. ** indicates ShopBot found the product but did not describe the price. * indicates that ShopBot failed to find the product even though the vendor was selling it.

Empirical Results(3)

1. Generality across product domains
   - CD domain (8 shops)
   - 1 day’s work on describing the CD domain
   - Shop successfully at 4 CD stores
   - BargainFinder(hand-crafted) currently shops successfully at 3 stores

Conclusion

1. ShopBot
   - Advantage
     - Find better prices in dramatically less time
     - Scale to multiple stores and product domains
   - Disadvantage
     - Not able to distinguish between upgrades to a product and product itself - need to do a more detailed analysis of product descriptions
     - Relies on a very strong bias
Conclusion

Future works
- Shopbot needs to be able to navigate a hierarchical organization.
- Shopbot should be able to run Java applets and attempt to analyze their output.

Appendix

The Shopbot Research Prototype was retired in 1998.
- It was a WWW shopping agent that enabled you to shop for computer software or CD's by manufacturer or title and artist queries.
- This technology was adapted in private industry and contributed to the development of an even better shopping agent at www.jango.excite.com.

Automatic ontology generator

Ontology
- A set of some terminology or keywords

Learning mechanism for tables

<table>
<thead>
<tr>
<th>Column</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColumnA</td>
<td>ValueA1</td>
<td>ValueA2</td>
<td>ValueA3</td>
<td>ValueA4</td>
</tr>
<tr>
<td>ColumnB</td>
<td>ValueB1</td>
<td>ValueB2</td>
<td>ValueB3</td>
<td>ValueB4</td>
</tr>
</tbody>
</table>

Learning mechanism for lists

1. Obtain datetime (2019/12/31 12:00:00 PM)
2. Insert into database
3. Update weather information
4. Display updated weather information
5. Display updated weather information
We have to find $\varepsilon$ in result page

Step 1: Save the resulting page for the sample query into a file
Step 2: The page is broken down into some logical lines
Step 3: Analyze each logical line to recognize its meaning and expresses it by a number
Step 4: Entire page is expressed by a sequence of numbers. Then find a repeated pattern in this sequence.

Figure 3: A HTML source and the portion

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Title</td>
<td>0</td>
</tr>
<tr>
<td>Condition</td>
<td>2</td>
</tr>
<tr>
<td>Price</td>
<td>0</td>
</tr>
<tr>
<td>Average Customer Review</td>
<td>5</td>
</tr>
<tr>
<td>Additional Info</td>
<td>0</td>
</tr>
<tr>
<td>Price</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 7: A HTML source and the portion