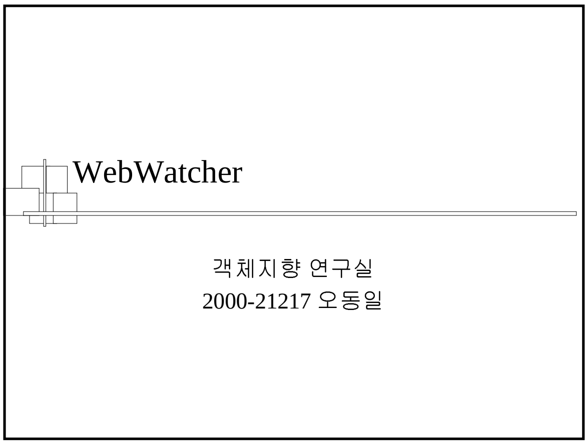
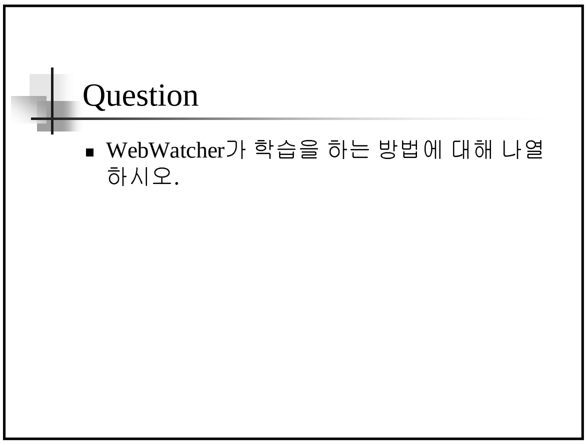


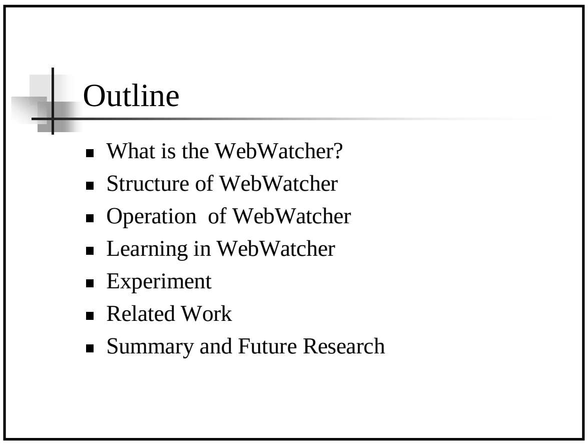
WebWatcher

객체지향 연구실
2000-21217 오동일



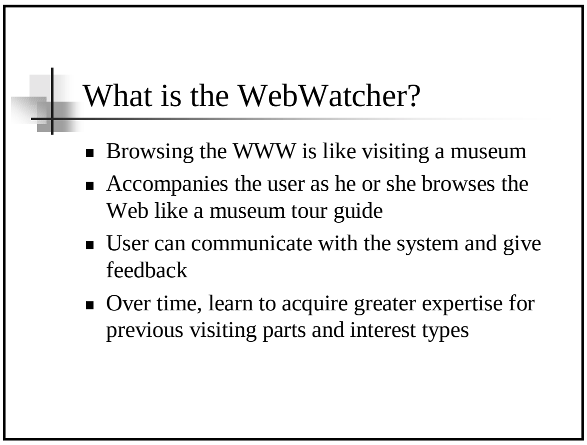
Question

- WebWatcher가 학습을 하는 방법에 대해 나열하시오.



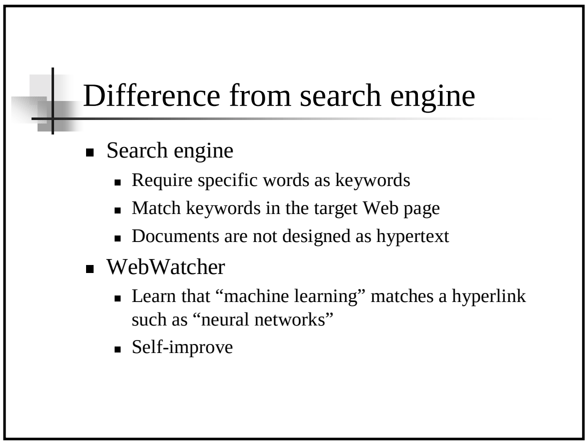
Outline

- What is the WebWatcher?
- Structure of WebWatcher
- Operation of WebWatcher
- Learning in WebWatcher
- Experiment
- Related Work
- Summary and Future Research



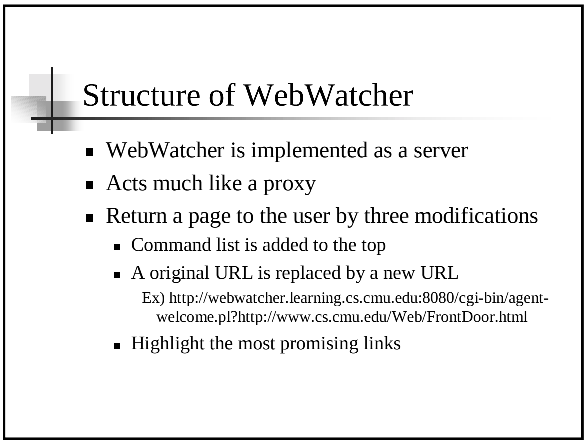
What is the WebWatcher?

- Browsing the WWW is like visiting a museum
- Accompanies the user as he or she browses the Web like a museum tour guide
- User can communicate with the system and give feedback
- Over time, learn to acquire greater expertise for previous visiting parts and interest types



Difference from search engine

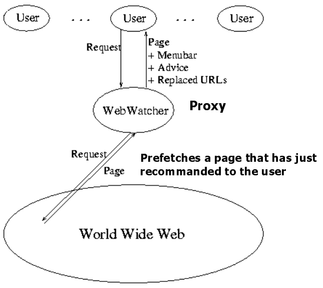
- Search engine
  - Require specific words as keywords
  - Match keywords in the target Web page
  - Documents are not designed as hypertext
- WebWatcher
  - Learn that “machine learning” matches a hyperlink such as “neural networks”
  - Self-improve



Structure of WebWatcher

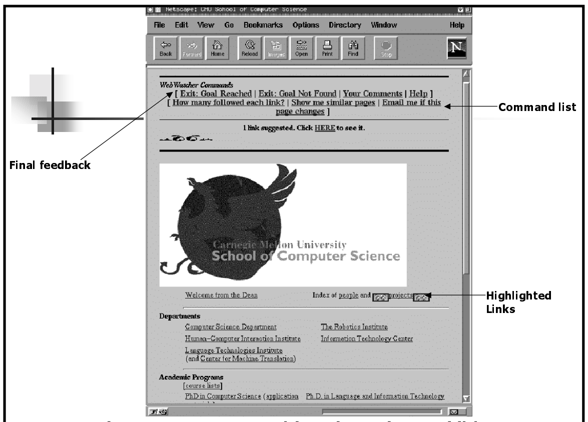
- WebWatcher is implemented as a server
- Acts much like a proxy
- Return a page to the user by three modifications
  - Command list is added to the top
  - A original URL is replaced by a new URL
    Ex) http://webwatcher.learning.cs.cmu.edu:8080/cgi-bin/agent-welcome.pl?http://www.cs.cmu.edu/Web/FrontDoor.html
  - Highlight the most promising links

Continued..



Operation of WebWatcher

- Invoked by clicking on the hyperlink “The WebWatcher Tour Guide”
- Leads us to a page of our current interest
- Accompanies us from the additions
  - WebWatcher Commands : communication method
  - Highlighted hyperlinks : suggestion directions
- To end the tour, two options in the command list
  - Exit : Goal reached
  - Exit : Goal not found



WebWatcher Commands

[Exit: Goal Reached | Exit: Goal Not Found | Your Comments | Help]
[How many followed each link? | Show me similar pages | Email me if this page changes]

1 link suggested. Click HERE to see it.

Carnegie Mellon University
School of Computer Science

Welcome from the Dean   Index of people and projects

Departments

Computer Science Department   The Robotics Institute
Human-Computer Interaction Institute   Information Technology Center
Language Technologies Institute
(and Center for Machine Translation)

Academic Programs
[course lists]
PhD in Computer Science (application   Ph.D. in Language and Information Technology

The Front-Door page with WebWatcher’s additions

Learning in WebWatcher

- LinkQuality
  - The probability that a user will select *Link* given the current *Page* and *Interest*
  - $LinkQuality : Page \times\ Interest \times\ Link \to [0,1]$
- Three approaches to learning this target function
  - Learning from Previous Tours
  - Reinforce learning
  - Combined method

Learning from Previous Tours

- Annotating each hyperlink with the interest on previous tours.
- Compare current user’s interest with descriptions of hyperlinks
- Interests and hyperlink description are represented by very high-dimensional feature vectors
- Elements of a vector are calculated using the TFIDF heuristic
- Similarity is calculated as the cosine between vectors

Continued..

- The value of *LinkQuality* for each hyperlink is the average similarity of the *k* keyword sets for this hyperlink
- Suggested if *LinkQuality* is above a threshold
- Maximum number of Suggested hyperlink is three

Keyword Vectoring

- Removes the suffix of the words
- Filter all commonly-used words
- Each word is weighted by TFIDF measure
- Term Frequency Inverse Document Frequency
  - $V_i = \Pi_c \times \text{Freq}(\text{Word}_i) \times [\log_2(n) - \log_2(\text{DocFreq}(\text{word}_i))]$
- Collection of documents is the set of all weighted keyword vectors

Learning from Hypertext Structure

- Augments a given hyperlink using words encountered in pages downstream of it
- Reinforcement Learning
  - Learns control strategies that select optimal actions in certain settings
  - The objective is to find paths through the Web which maximize the amount of relevant information

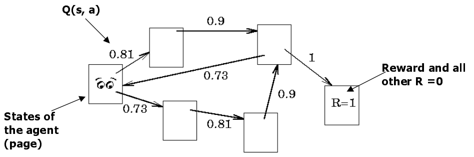
Continued..

- Discounted sum of future rewards

$Q(s_t, a) = \sum_{i=0}^{\infty} \gamma^i \cdot R(s_{t+1+i})$ **The goodness of an action a**

S_t : state in at time t γ : discount factor($0 \leq \gamma < 1$)

R(s): reward a : action

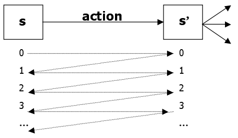


Example state space (γ=0.9)

Continued..

- Under certain conditions, Q function can be iteratively approximated by updating estimate for Q(s, a) repeatedly

$$Q_{n+1}(s, a) = R(s') + \gamma \max_{a' \in actions\_in\_s'} [Q_n(s', a')]$$



Continued..

- Reinforcement Learning and Hypertext
  - Interest "*intelligent*"
    - $R_{intelligent}(s)$: the TFIDF value of "*intelligent*" for page s
    - $Q_{intelligent}(s, a)$: the sum of discounted TFIDF values of "*intelligent*" over the optimal tour beginning with a
  - WebWatcher uses a separate reward function $R_w(s)$ and learns a distinct $Q_w(s, a)$ for every word w
  - Problem
    - WebWatcher cannot expect that users will always stick to pages it has already seen

⇨ **Distance-weighted 3-nearest neighbor function approximator**

Experiment

	Accuracy
Random	31.3%
Popularity	41.9%
Match	40.5%
Annotate	42.2%
RL	44.6%
Combine	48.9%

Random : suggests hyperlinks at random from current page

Popularity: followed most frequently in the past

Match: TFIDF-cosine similarity between underlined text and user's interest

Annotate: Learning from Previous Tours

RL: Reinforcement learning

Combine: combines above all using logistic regression

Experiment

	Accuracy
Random	22.4%
Annotate	42.9%
Human	47.5%

Comparison to human performance on the Front-Door page

Related Work

- Letizia
- Syskill and Webert
  - Manually constructed index page
  - User can rate hyperlinks off page
  - System use the ratings to learn
- Lira
  - Works in an offline setting

Conclusion and Future Research

- WebWatcher is a Self-improving tour guide agents
- Provide helpful advice to many users
- But there are some topics for future research
  - Personalized WebWatcher
  - Combining user-specific and Web locale-specific learning
  - Richer dialogs with users
  - New machine learning algorithms
  - Intelligent distributed hyperlinks