

CS556 Distributed Pervasive Computing Presentation-1 Summary

An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web.

Pradeep Bhukya

pbhukya@mix.wvu.edu

Introduction:

World Wide Web is vast and dynamic in nature and there is need for flexible information finding systems. One of the major challenges is constantly monitoring the web for up to date relevant information. The solution is creating a customizable, intelligent interface to the World Wide Web that assists the user in locating the specific information. User interface should be customizable by users for their own purposes. User interface should constantly monitor the network without continuous inputs from the user. User interface should be customizable by users for their own purposes. WAWA (Wisconsin adaptive web assistant) takes instructions from the user and looks for the specified info. Wawa allows users to create assistants to constantly monitor the network. User created and machine learned training examples change the behavior of the system. User provides information in a special language which is converted into two neural networks score link, score page. These two scoring functions search the web and determine which pages to collect for the user.

Approach Taken:

The Score This Link and Score this page functions are programmed using the user's advice instructions in advice language. Initially the user instructions are converted by the kbann algorithm as neural network that execute the user's advice.

WAWA Algorithm:

- Unless they have been saved to disk in a previous session, create the *Score Link* and *Score Page* neural networks by reading the user's initial advice (if any).
- Either
 - (a) Start by adding the user-provided URLs to the search queue; or
 - (b) Initialize the search queue with URLS that will query the user's chosen set of web search-engine sites.

The user can also specify a depth limit that puts an upper bound on the distance the system will navigate from the initial *URLs*.

- **Execute the following concurrent processes.**

- ***Independent Process #1***

While the search queue is not empty nor the maximum number of *URLs* visited,

Let *URL to Visit* = *pop* (*search queue*).

Fetch *URL to Visit*.

Evaluate *URL to Visit* using *Score Page*.

If score is high enough, insert *URL to Visit* into the sorted list of best pages found.

Use the score of *URL to Visit* to improve the predictions of the *Score Link* function.

Evaluate the hyperlinks in *URL to Visit* using *Score Link* (however, only score those links that have not yet been visited this session). Insert these new *URLs* into the (sorted) search queue if they fit within its max-length limitation.

- ***Independent Process #2***

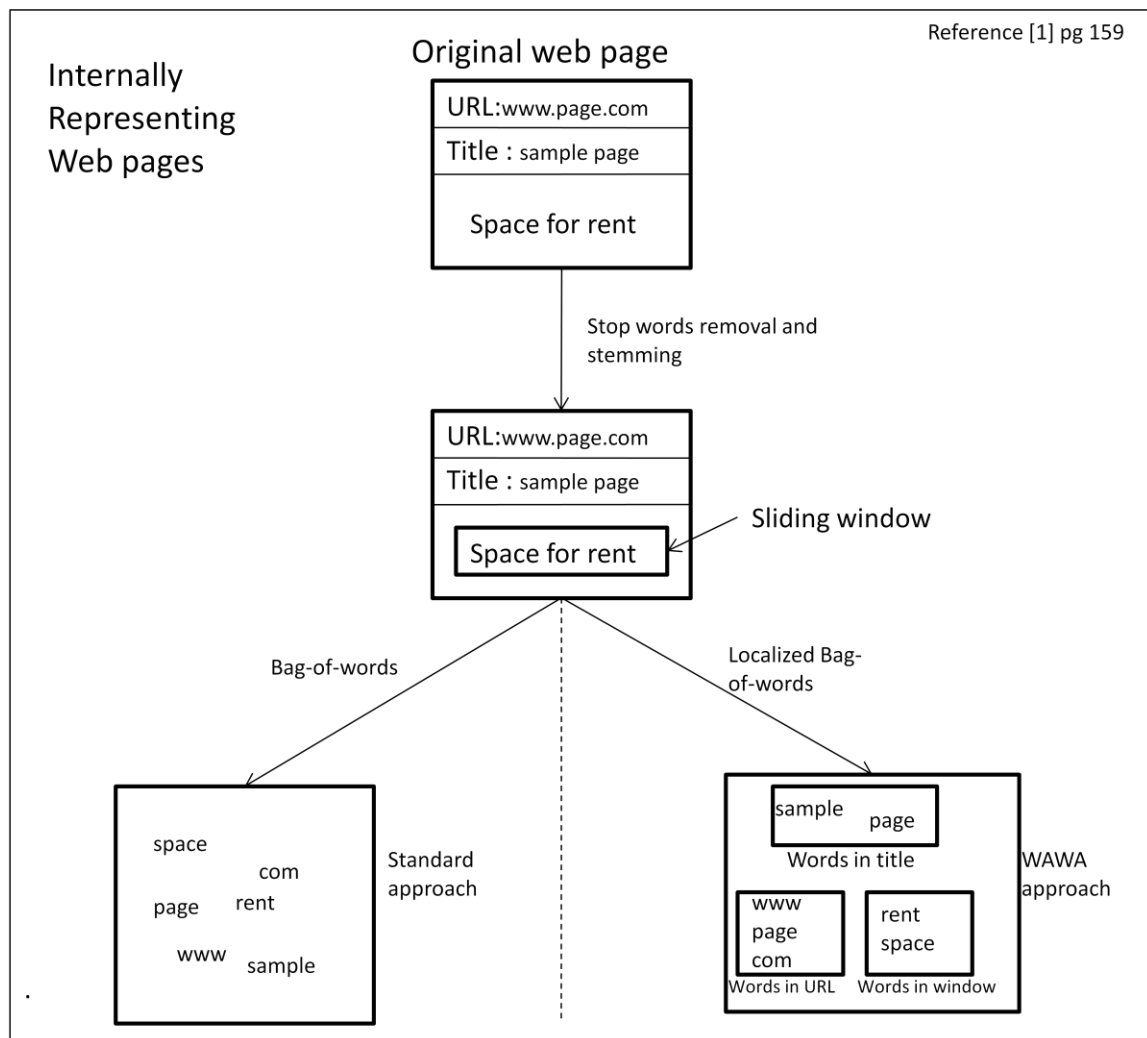
Whenever the user provides additional advice, add it to the appropriate neural network.

- ***Independent Process #3***

Whenever the person rates a fetched page, use this rating to create a training example for the *Score Page* neural network.

Rating the web page:

WAWA extracts the features and primitives from the user's advice instruction. Removal of 'stop' words and stemming all words ('walked' becomes 'walk') is performed on the fetched web page. WAWA slides a fixed width (15 words) window across a web page. The *Score Page* network scores the page by sliding across the web page. Similarly the *Score page* slides over the hypertext of the hyper link for scoring. Use of a richer representation called localized bag-of-words model to preserve the order of words and to calculate the number of occurrences of the words in the page and the distance between the word's next occurrences. This approach also has Localized bag-of-words for words in title, words in *URLs* and words in sliding window.



WAWA also considers other input features like first and last N words in the localized bag-of-words, length and date of creation of the page, sliding window location, word bag size and no of words mentioned in the advice present in the bag of words.

Sample of the advice language instruction:

WHEN consecutive Words In Hypertext

(Intelligent user interface)

STRONGLY SUGGEST FOLLOWING HYPERLINK

Future Work:

Current efforts are being made on developing easy to use interface for composing advice. Like menu based design that converts the user's choice into instruction in WAWA's advice language.

References:

1. An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World-Wide Web **Jude Shavlik, Susan Calcari, Tina Eliassi-Rad, and Jack Solock.**
2. E. Brill. Some advances in rule-based part of speech tagging. In *Proc. National Conf. on AI (AAAI-94)*, pages 722-727, 1994.
3. T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the World Wide Web. In *Proc. Intl. Joint Conf. on AI*, pages 770-775, 1997.
4. H. Lieberman. Letzia: An agent that assists web browsing. In *Proc. Intl. Joint Conf. on AI*, pages 924-929, 1995.
5. R. Maclin and J. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22:251-281, 1996.
6. G. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38:39-41, 1995.
7. M. Pazzani, J. Muramatsu, and D. Billsus. Identifying interesting web sites. In *Proc. National Conf. on AI (AAAI-96)*, pages 54-61, 1996.
8. D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323~533-536, 1986.
9. G. Salton. Developments in automatic text retrieval. *Science*, 253:974-979, 1991.
10. T. Sejnowski and C. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145-168, 1987.
11. J. Shavlik and T. Eliassi-Rad. Intelligent agents for Web-based tasks: An advice-taking approach. In *AAAI/ICML Workshop on Learning for Text Categorization*, 1998.
12. R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9-44, 1988.
13. G. Towell and J. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70:119-165, 1994. 160