Creating Topic-Specific Automatic Multimodal Presentation Mining the World Wide Web Information

Shaikh Mostafa Al Masum, Mitsuru Ishizuka, Md. Tawhidul Islam[†]
Dept. of Information and Communication Eng., University of Tokyo, 7-3-1 Hongo, Bunkyo Ku, Tokyo, Japan
[†] Micros-Fidelio Australia Pvt. Ltd., 13, Narabang Way, Belrose, NSW, Australia.
mostafa_masum@ieee.org, ishizuka@miv.t.u-tokyo.ac.jp, mislam@micros.com

Abstract

The paper describes the integration between web intelligence and character-based software agent manipulation with the notion of autonomous information services. The system, 'Auto-Presentation', builds a presentation automatically by parsing, summarizing and correlating information collected from the Internet based knowledge sources after receiving the presentation topic from the user. The system, with the help of a group of character based software-agents, presents automatically the retrieved information about the topic verbally with accompanied slides, different gestures and affects associated with presenter (e.g. the character agents). With a brief literature re-view, in section 1 the basic idea of the system is explained. Section 2 describes the architecture and explains different components of 'Auto-Presentation'. Section 3 describes necessary algorithms. Section 4 depicts some test results and evaluations. Section 5 concludes the paper with the trail of future work.

Keywords: Web Presentation, Web Data-Mining, Search Engine, Autonomous Information, Agents

I. INTRODUCTION

Internet is the biggest online multi-disciplinary information repository in the world. Due to availability of large quantity of data and the dynamic nature of web pages, the task of information retrieval is becoming more challenging. User interested in a certain topic can exploit many information resources of various nature, content and characteristics. Hence the idea of building and presenting an automatic multimodal presentation of a particular topic or query could be thought as a new dimension of autonomous information service and next generation of web search engines. The developed system, Auto-Presentation, is an attempt towards this notion. In the system several issues of web intelligence blended with text processing and scripting of character based software agents have been incorporated. So the research is encircling different research outcomes like html parsing; web page search, extraction and summarization; question answering system; information retrieval and agents' markup language for scripting affects and gestures of character based agents and adopting some extensions and modification of the above topics.

A. Related Literature

Web pages are often very "noisy" in the sense that they might contain many unrelated information. So, many unrelated text segments may be identified by an HTML-parser. There any many HTML-parses [17][18] that can parse an HTML page by tagging and extract contents from different section of the document, but the limitation of those parser for our purpose is that, as web pages may emphasize phrases or long text segments unrelated to the key information, further parsing is required to extract the important text and concept from a document. Hence the system, Auto-Presentation, employs an HTML-parser that outputs data in the form of heading and associated text related to heading where both heading and text segment are co-related with the main topic of the presentation.

Web searching, extracting and finally summarizing useful information are active research areas since last decade. In brief, the techniques include Keyword-based search (e.g., [1][2]), Web queries, Wrapper Induction for Information, Effective Resource Discovery, User Preference-based search and Content or Context based[13] summarization. Keyword-based search using search engines like Google, Yahoo, Wikipedia and AltaVista is taken as the initial step to collect the links of potential information relevant to presentation topic. Web query languages allow the user to retrieve data (e.g. table to MS Excel file) from web pages by using extended database query languages. This is not required for our problem. Wrapper Induction for Information approaches (e.g., [3]) are not also suitable because a Wrapper is a procedure for extracting tuples from a particular information source. Hence, they are not designed for finding significant concepts and exploratory texts associated with the different concepts of user-specified topics. Effective (Web) Resource Discovery aims to find Web pages relevant to users' requests or interests (e.g., [5]). This approach uses techniques such as link analysis, link topologies, and text classification methods to find relevant pages. However, relevant pages, which are often grouped by keywords, are well enough for our purpose because we need to further route the contents of the Web pages to discover presentation headings/sub-heading of the topic and descriptive information associated with those headings. In the user preference approach, information is presented to the user according to his/her preference specifications and this is not helpful for our problem. Content based summary utilizes textual content of the web documents in

question. The disadvantage of this method becomes evident when a particular web page contains little textual content and relies mostly on visual language communication. Context-based method [6], which are making use of the hypertext structure of the web, exploit the paragraphs or other text units that are close to the links pointing to the particular document are used to create the. For our approach we used the mixed approach, both content and context based, depending on some conditions.

Related work to ours is question-answering (e.g., [7], [8][14]). A question-answering system is used to answer user questions by consulting a repository of documents. [7] utilizes the snippet returned from a search engine to help find answers to a question. Potential web pages related to topic are enlisted in the similar approach to answering questions. We have incorporated some of the heuristics from question-answering research to finding such informative pages and also utilize some of the concepts of [8] which explained about mining topic-specific concepts and definitions collected from web pages. However, the total task is different in terms of building presentation outline dynamically and associating of summarized text chunks and images to the related heading and finally presenting the presentation by some visual character agents with some soft of affective support. We also make use of the web presentation characteristics, web-based encyclopedias, web page structures as clues to process user requests to make presentation.

To cite some of the more prominent applications, embodied characters are now used as virtual tutors in interactive learning environments [15], as virtual sales agents and presenters [9], and as virtual actors for entertainment as well. Recent years show a growing interest in animated characters to enhance learning in computer-based interactive learning environments [9]. Admitting this we also devised the mechanism of generating scripts automatically for the character agents (Microsoft Agent [11]) to act accordingly. The Multimodal Presentation Markup Language, MPML [9, 10], has been used to script the agents.

B. Concept of '*Auto-Presentation*'

The objective of the system task is to help the user learn on the Web like attending a seminar or talk. In the system, we do not require extensive level of linguistic analysis or learning rather than shallow language processing. In the context and content of the Web, we rely on conventional search engines, web encyclopedia and exploit the structure of the web pages to identify candidate phrases for information retrieval. To build the presentation first web encyclopedia is consulted and then for more information, we approach to multiple but unique web pages. Using template based (explained in section 2) data mining technique, the system is able to associate and co-relate text segments related to outline of the presentation. Moreover the technique integrates the technologies of finding and building presentation outline, salient text finding and associating with relevant outlines, Image retrieval, to help the user to perform systematic understanding of a topic explained by character-based agents. The core features of 'Auto-Presentation' are:

- Understand the user request for the topic to present
- Interactive character agents
- Dynamic building of presentation outline
- Web Search, Filter, Extract, Rank, Summarize and Associate Text to Outline
- Dynamic creation of MPML [10] script for agent scripting with affective support
- Implementation of Microsoft Agent System [11]

II. SYSTEM ARCHITECTURE

Our system, *Auto-Presentation*, is consisting of multiple agents performing specific tasks. Fig. 1 shows the architecture of the system in terms of agent interaction.
The names of the agents are self-explanatory. The actions of the above agents conform to the core features mentioned above. The interaction of the agents would be well understood by consulting the functional flow diagram of the system as indicated in fig. 2.

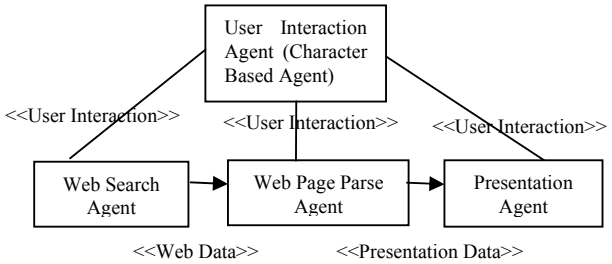

Fig. 1 Multi-Agent architecture of the system

III. NECESSARY ALGORITHMS

First overall algorithm of the total operation of the system is described. To explain the algorithm we admit a heuristic that if a search topic is found in online encyclopedia (e.g. Wikipedia in this case), the retrieved information from the encyclopedia can be considered as well structured and hence the initial outline of the presentation can be instantiated after the data and information structure retrieved from encyclopedia but if online encyclopedia failed to retrieve significant information we extract data from the web-pages based on the template as described in the table I.
Hence, the overall algorithm follows,

Begin

Load_Agent (*List_Of_MSAgents*)

Instruct_Agent_To_Interact (*Context*)

Topic= Analyze_User_Query (*queryString*);

urlWiki = Search_Wikipedia(*Topic*)

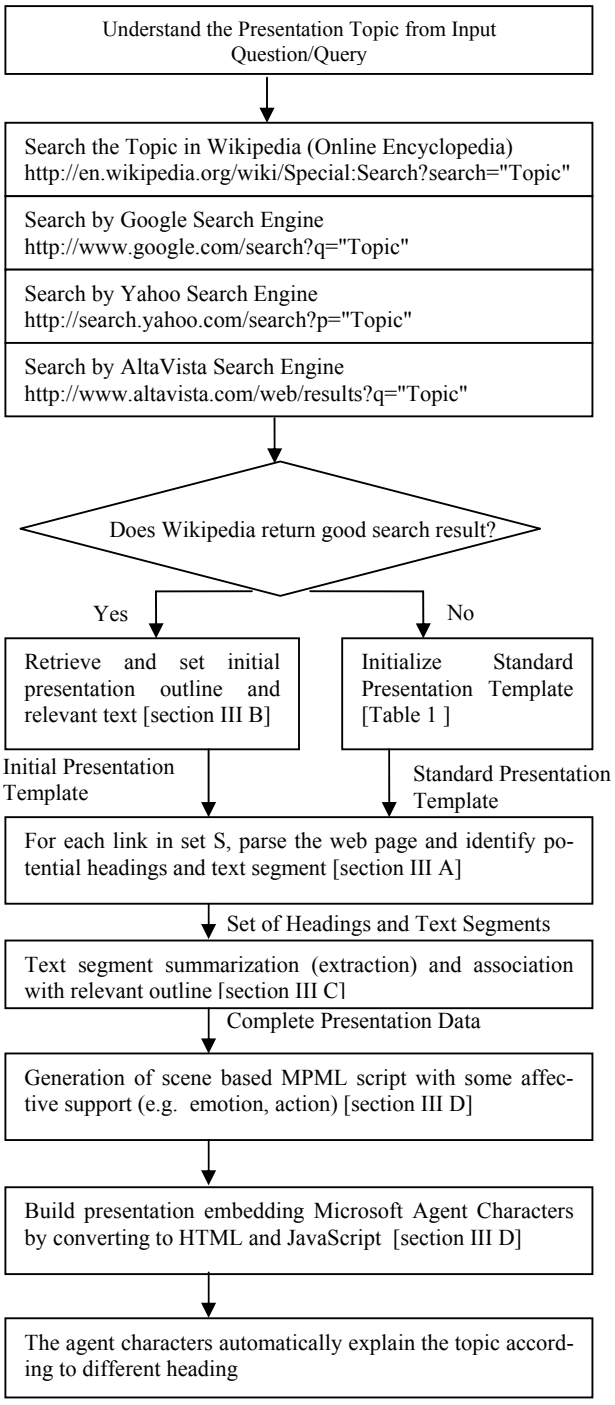

Fig. 2 Sequence of operation of the system

urlsGoogle = Search_Google (*Topic*) //Top Ten

urlsYahoo = Search_Yahoo (*Topic*) //Top Ten

urlsAltaVista = Search_AltaVista (*Topic*) //Top Ten

urlsImage = Search_Google_For_Images (*Topic*)

uniqueUrlSet = makeUniqueUrlSet (*urlsGoogle*, *urlsYahoo*, *urlsAltaVista*)

If *urlWiki* not returns 'Badly formed search query' then

Parse the web-page returned by *urlWiki* to Extract Initial Outline, Text and Images

PT = Set_Initial_Presentation_Template

Else

PT= Initialize_Standard_Presentation_Template

For each link, *WP*, in *uniqueUrlSet* do

Plain_Page = Plain_Parse (*WP*)

Extracted_Page = Extract_Data (*Plain_Page*)

Outlines = Find_Closeness_In_Exisiting_Presentation_Template (*Extracted_Page*)

For each retrieved outline in *Outlines* do

If outline is significantly close to existing one in *PT* then Begin

Text_Related_To_Outline = Get_Associated_Text (*Extracted_Page*)

Extracted_Text = Extract_Ranked_Text (*Text_Related_To_Outline*)

Closeness_Factor = Measure_Closeness (*Extracted_Text*, *Previously_Added_Text*)

If *Closeness_Factor* is within the Threshold then

No need to add the Text in the Presentation Template

Else

Add the *Extracted_Text* to the Presentation Template

End

Else

Begin

New_Heading = Generate_New_Heading (*Outlines*)

Text_Related_To_Outline = Get_Associated_Text (*Extracted_Page*)

Extracted_Text = Extract_Ranked_Text (*Text_Related_To_Outline*)

Add_Heading_To_Outline (*PT*, *New_Heading*)

Add the *Extracted_Text* to the Presentation Template

End

Next outline

Next link

MPML_Script = Make_MPML_Script (*PT*)

Auto_Presentation = Convert_To_HTML_JavaScript (*MPML_Script*)

Load_Presentation_In_Web_Container (*Auto_Presentation*)

End

In the next sub-sections we explain the necessary algorithms more details. The comparisons of other or justification of given algorithms are not discussed here.

Table I Format of Standard Presentation Template

Title	**Key/Cue Phrase to mine around the text**
What/Who is [Topic]	about us, about [Topic], introduction, mission, objective
Whereabouts [Topic]	contact us, profile, location, services
Why [Topic]	The text snippet retuned by search engines
How [Topic]	The snippet returned by search engines
The short text not already inserted in present template and found in between emphasizing tags like: <h1>,...,<h4> <b> <big> <i> <u> <dt>	The text following the emphasizing tags and the sentences that give significant sentence selection score according to equation 3.

A. Web-Page Parsing Algorithm

For each page (in the set of unique link list) do

1. Read a line until end of file
2. If the line is between <body> </body> tag then
   a. Retrieve text between emphasizing tags like: <h1>,...,<h4> <b> <big> <i> <u> <dt>. These are the prominent candidates for outline.
   b. Ignore the text that contains an URL or an email address, terms related to a publication (e.g. journal, conference, and proceedings), an image between the markup tags.
   c. Ignore the text which is too long (more than 125 words in a line).
   d. Strip tag from that line.
   e. Remove unnecessary characters, redundant white spaces
   f. Retrieve links of images (if there is any)
   g. Retrieve the potential hyperlinks (e.g. about, contact, mission, more info etc.) from the line (if any) to explore further information
   h. Write the line to the File.

Output from each parsed page: The list of tuples of potential headings and text chunk, The List of images and expected hyperlinks.

B. Presentation Object Generation Algorithm

1. Retrieve the page returned by Wikipedia
2. If the page doesn't contain 'Badly formed search query' term, it indicates Wikipedia has some significant and structured information. Else goto step 5
3. Parse the Wikipedia page according to algorithm [section 3A]
4. Initialize presentation template (outline and associated text) using information from Wikipedia. Goto step 6.
5. Instantiate standard presentation template (as shown in table I)
6. For each parsed page do
7. Get heading(s) and associated bulk text tuples
8. If the size of heading is more than one then create new heading by ranking in terms of frequent word and keyword else consider the single head.
9. Measure closeness of the heading with the other headings inserted already in the presentation template
10. If the closeness is non-negative number (positive number indicates a close match to an existing heading in the outline), extract and associate text using algorithm in section 3C
11. Else add the heading and extracted text in the template
12. If the heading doesn't associate with text, retrieve information from other hyperlinked page of link (using the heuristics mentioned in table I)
13. For each heading try to match an image from the list of images retrieved by Google's image search and Wikipedia by considering the source and name of the image file.
14. The maximum number of presentation heading is kept limited to 25 (i.e. maximum 25 slides to show)

Output: A presentation object containing presentation outline and associated text chunks and images and necessary references.

C. Algorithm to Summarize (Extract) Text

For summarization almost a similar formula to Average TF-IDF [12],[16] has been used to measure the relevance score of the sentences associated with heading(s). The Avg-TF-ISF [12] has been calculated for each sentence. Then a specific percentage is multiplied with this value. And then a portion of the total word count of that sentence is calculated. This is done by multiplying the total word count of that sentence with a percentage value. These two values are added to get the relevancy score of that sentence. Lastly those sentences that have a value above the specified percentage of the maximum relevance score (Summary Threshold) are selected for creating a summary/extract for that heading. Here both the Avg-TF-ISF and the word count of the sentence contribute to the measurement of the sentence relevancy. The rule to calculate Avg-TF-ISF is given as below:

$$TF\text{-}ISF(w,s) = TF(w)*(1+\log(|S|/SF(w) \quad \ldots\ldots\ldots\ldots (1)$$

$$Avg\text{-}TF\text{-}ISF(s) = \sum TF\text{-}ISF(w,s)/W(s) \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where,

$TF(w,s)$ = The frequency of word *w* in Sentence *s*

|S| = Total number of Sentences

SF(w)= Number of Sentences the Keyword *w* was found

W(s) = Number of words in the Sentence, s

Avg-TF-ISF(s) = the score of the sentence from its words' TF-ISF

This is an adaptation of the conventional TF-IDF formula.

To calculate the relevance score of each sentence we used:

Relevance score(s) = Avg-TF-ISF * TF-ISF Percentage + W(**s**) * WordPercentage ……..…… (3)

To select sentences we have:

IF Relevance score(**s**)>Max^m Relevance Score * SummaryThreshold

THEN Sentence ***s*** is selected for summary/extract.

The algorithm as follows:

1. If there are previously any text-chunk collected for the heading, add those text chunks with the present text chunk to summarize for better performance.
2. Using a shallow language parser eliminate high frequency words and do other pre-processing
3. For each sentence
   i. Calculate the TF-ISF for each word with respect to heading text and search engine retrieved text snippet
   ii. Take the Average of the TF-ISF of those words. This is Avg-TF-ISF or score of that page
   iii. To calculate the relevance score use

Relevance score(s)=Avg-TF-ISF* TF-ISFPercentage + W(s)*WordPercentage

4. Find the Maximum Relevance score
5. Select those sentences whose score is above than Max^m Relevance Score*SummaryThreshold
6. Insert the selected sentences, to associate it with the corresponding heading, in the presentation template.

Output: Extracted text associated with heading and topic.

D. Algorithm to Build and Play Presentation

Input: Presentation object containing presentation outline and data.

The algorithm as follows:

1. From the presentation object create HTML files corresponding to outlines, each heading.
2. Generate scripts using Multimodal Presentation Markup Language as follow
   a. Each heading is considered as a scene to be acted by two MS Agent character
   b. According to the agent's role (reading or listening) select the necessary affects of the agent and configure the agent's tone, agreeableness, activity (for detail see [9])
   c. Select the important lines to be spoken by the agent (if the text is too much to be spoken)
   d. Generate necessary MPML tags to control the agents behaviors
3. Save the MPML script to file.
4. After creating all the necessary scenes convert the MPML scripts to HTML and JavaScript using a converter module (the algorithm for conversion is not in the scope of this paper) and Finally make a HTML documents (index.html) with several frames to load presentation files in respective frames.
5. The Java Scripts does the necessary automation for presentation with the help of several MS Agent characters.

IV. TEST AND EVALUATION

In order to test the system we recorded the execution time in terms of time taken to make a presentation (not included here due to space limitation) and to evaluate usability of the system we interviewed 25 students to test the system and asked them to fill up a questionnaire. In table II we present some evaluations which indicate their assessment for the automatic presentation In future we are planning to perform more extensive tests by deploying the application on a web-server.

Table II Comments of some students

No	Question Asked	Did It Work	Quality of Data Presented	Overall Quality
1	Tell me about love	Yes	Very good	Fine
2	Tell me about Hell	Yes	Very Good	Acceptable
3	What is Big Bang?	Yes	Very good	Acceptable
4	Tell me about Formula 1	Yes	Very Good	Acceptable
5	Tell me about Coral Reef?	No	Not Good	Not Good
6	What is Heaven?	Yes	Very good	Good
7	Tell me about F22	No	Not Good	Not Good
8	What is J2EE	Yes	Good	Good
9	What do you know about AI?	Yes	Very Good	Fine
10	What can you tell about Pope?	Yes	Very Good	Fine
11	What is life	Yes	Good	Fair

V. CONCLUSION

It is obvious that this smart and intelligent agents' interactive auto-presentation is a robust approach for information retrieval and learning from web. The proposed system's behavior is different from that of conventional information retrieval systems (e.g. [7], [8], [13]) in several aspects. First one is the agent interaction (see snap 1, top right) that always keep a user aware of the system's status (so the user doesn't feel bore while doing background processing). Secondly task-oriented, semi-autonomous and collaborative multi-agent architecture emphasizes on some emotive support by scripting some emotive tags by MPML to make the presentation more live (see snap 1) and finally a quick concept building approach around the topic has been implemented by considering presentation templates to be filled out by the information miner.

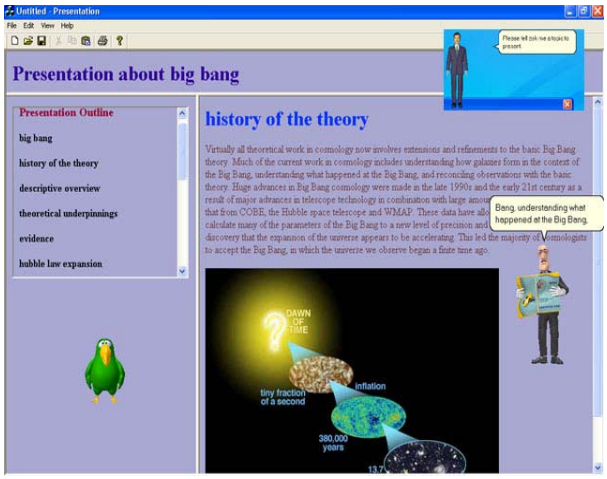

Fig. 3 Snapshot of a Sample Automatic Presentation

For developing the software we used MS Visual C++, Microsoft Speech API and Microsoft Agent APIs [11]. We admit that additional work is necessary to optimize the system so that it can support high loads with fast response and multi-user support. Some fine tune is also required for information retrieval and for this we are concentrating the structure of web documents. We are also revising the MPML model to support bi-directional emotional support to make the presentation more live and user focused.

REFERENCES

[1] Brin, S., and Page, L., "The anatomy of a large-scale hypertextual Web search engine", *Computer Networks and ISDN Systems*, 1998, 30(1–7):107–117.

[2] A. Rakhshan, L. B. Holder and D. J. Cook, "Structural Web Search Engine", International Journal of Artificial Intelligence Tools, 13(1), pages 27-33, 2004

[3] Cohen W., Fan W., "Learning page-independent heuristics for extracting data from Web pages", In Proc. of WWW8, 1999

[4] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. & Nevill-Mainning, C.G., "Domain-specific keyphrase extraction" , In Proc. of 16th IJCAI, 1999

[5] Dean, J. & Henzinger, M.R., "Finding related pages in the World Wide Web", In Proc. of WWW8, 1999

[6] Amitay, E., and Paris, C., "Automatically summarizing web sites: is there any way around it?" Proc. of the 9th International Conference on Information and Knowledge Management (McLean, Virginia, November 2000), 173-179.

[7] Kwok, C., Etzioni, O. & Weld, D.S., "Scaling question answering to the Web", In Proc. of WWW10, 2001 http://mulder.cx/search.html

[8] Liu B., Chin C. W., and Ng, H. T., "Mining Topic-Specific Concepts and Definitions on the Web", In Proceedings of the Twelfth International World Wide Web Conference (WWW'03), Budapest, Hungary, 2003

[9] Helmut P., Sylvain D., and Mitsuru I., "Scripting Affective Communication with Life-like Characters in Web-based Interaction Systems", *Applied Artificial Intelligence*, Vol.16, Nrs.7--8, pp.519--553, 2002

[10] M. Ishizuka, T. Tsutsui, S. Saeyor, H. Dohi, Y. Zong, and H. Prendinger, "MPML: A multimodal presentation markup language with character control functions", In Proceedings Agents'2000 Workshop on Achieving Human-like Behavior in Interactive Animated Agents, pages 50-54, 2000.

[11] Microsoft® Agent (www.microsoft.com/msagent)

[12] Yihong Gong, Xin Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis", NEC USA, C & C Research Laboratories, USA, SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA

[13] Glover, E. J., Tsioutsiouliklis, K., Lawrance, S., Pennock, D. M., and Flake, G. W., "Using web structure for classifying and describing web pages", Proc. of 11th International WWW Conference (Honolulu, Hawaii, May 2002), 562-569.

[14] Cooper, R.J. & Rüger, S. M., "A simple question answering system", In Proc. of TREC 9, 2000

[15] W. L. Johnson, J. W. Rickel, and J. C. Lester, "Animated pedagogical agents: Face-to-face interaction in interactive learning environments", *International Journal of Artificial Intelligence in Education*, 11, 47-78, 2000.

[16] Liren Chen and Katia Sycara, "WebMate: A personal Agent for Browsing and Searching", Carnegia Mellon University, September 30, 1997, http://www.cs.cmu.edu/~softagents/webmate

[17] Monika R. Henzinger, "Algorithmic Challenges in Web Search Engines", Internet Math 1 (2003), no. 1, 115–123

[18] HTML Parser, http://htmlparser.sourceforge.net/