'*Auto-Presentation*': A Multi-Agent System for Building Automatic Multi-Modal Presentation of a Topic from World Wide Web Information

Shaikh Mostafa Al Masum
Department of Information and Communication Engineering
University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo,113-8656, Japan
mostafa_masum@ieee.org

Mitsuru Ishizuka
Department of Information and Communication Engineering
University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo, 113-8656, Japan
ishizuka@miv.t.u-tokyo.ac.jp

Md. Tawhidul Islam
Micros-Fidelio Australia Pvt. Ltd., 13, Narabang Way
Belrose, NSW 2085
Australia
mislam@micros.com

Abstract

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 the topic verbally with accompanied slides and different gestures. Section 1 provides brief introduction and 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.

1. Introduction

Internet is the biggest online multi-disciplinary information repository in the world. 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 method of information visualization for web searches. 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 with a notion to implement some extensions and modification of html parsing; web page search, extraction and summarization; question answering system; information retrieval and agents' markup language for scripting gestures as well as affects.

1.1. Concept of '*Auto-Presentation*'

In the system, extensive level of linguistic analysis or machine learning are not required rather than shallow language processing. For web contents, we rely on conventional search engines, web encyclopedia and exploit the structure of the web pages to identify candidate phrases for information retrieval, similar to [1]. To build the presentation first web encyclopedia is consulted and then for more information, we approach to unique web pages returned by several search engines. Using template based (explained in section 3) 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 and Image retrieval to help the user to 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 [2] script for agent scripting with affective support
- Implementation of Microsoft Agent System [3]

2. System Architecture

Figure 1 shows the multi-agent architecture of the system in terms of agents' interaction.

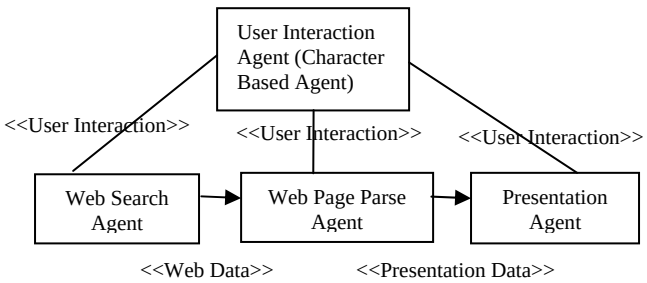

Figure1. Multi-Agent architecture of the system

The names of the agents are self-explanatory. The actions of the above agents conform to the core features mentioned in 1.1. The interaction of the agents would be well understood by consulting the data flow diagram of the system as indicated in figure 2

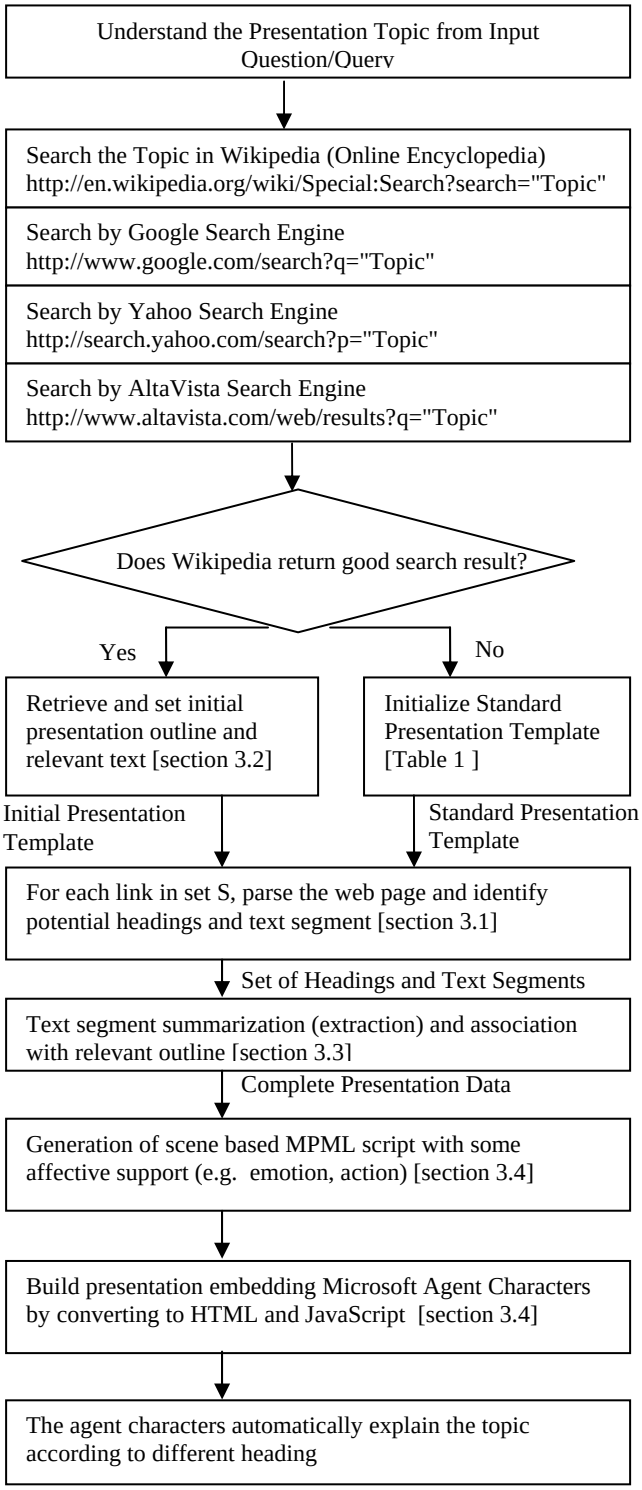

Figure 2. Data flow diagram of *'Auto-Presentation'*

3. 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 need a data-structure or template for that context. Hence the table 1 comes into picture to overcome such problem.

Table 1: 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.

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*)
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.

3.1. Algorithm for Web-Page parsing

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, Remove unnecessary characters, redundant white spaces
   e. Retrieve links of images (if there is any) and Retrieve the potential hyperlinks (e.g. about, contact, mission, more info etc.) from the line (if any) to explore further information

3.2. Algorithm to Generate Presentation Object

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 3.1]
4. Initialize presentation template (outline and associated text) using information from Wikipedia and Goto step 6.
5. Instantiate standard presentation template (as shown in table 1)
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. 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 3.3, Else add the heading and extracted text in the template
10. If the heading doesn't associate with text, retrieve information from other hyperlinked page of link (using the heuristics mentioned in table 1)
11. The maximum number of presentation heading is kept limited to 25.

3.3. Algorithm to Summarize (Extract) and Associate Text with Outline

To calculate the relevance score of each sentence we used:
Relevance score(s) = Avg-TF-ISF * TF-ISF Percentage + W(**s**) * WordPercentage
To select sentences we have:
IF Relevance score(**s**) > Max^{m} Relevance Score * SummaryThreshold
THEN Sentence ***s*** is selected for summary/extract.

3.4 Algorithm to Build Automatic 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 (MPML) as following,
    a. Each heading is considered as a scene to be acted by two MS Agent character[3]
    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 [2])
    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 behavior
3. 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). Java Scripts does the necessary automation for presentation.

4. 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) 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 2 we present 15 such evaluations which indicate their assessment for the automatic presentation.

Table 2: Comments from some students

Test 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	Tell me about Micros-Fidelio	Yes	Not Good	Not Good
4	What is Big Bang?	Yes	Very good	Acceptable
5	Tell me about Formula 1	Yes	Very Good	Acceptable
6	Tell me about Coral Reef?	No	Not Good	Not Good
7	What is Heaven?	Yes	Very good	Good
8	Tell me about F22	No	Not Good	Not Good
9	Tell Me about Top Gun	No	Not Good	Not Good
10	What is J2EE	Yes	Good	Good
11	What do you know about AI?	Yes	Very Good	Fine
12	Explain about Pole Shift.	Yes	Not Good	Acceptable
13	What can you tell about Pope?	Yes	Very Good	Fine
14	Where is University of Tokyo?	Yes	Good	Acceptable
15	What is Life?	Yes	Very Good	Fine

5. Conclusion

The proposed system's behavior is different from that of conventional information retrieval systems in several aspects. First one is the agent interaction that always keeps a user aware of the system's status. 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 and finally a quick concept building approach around the topic has been implemented by considering presentation template to be filled out by the information miner. Further refinements in the algorithms are necessary to improve information retrieval.

References

[1] 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

[2] 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

[3] Microsoft® Agent http://www.microsoft.com/msagent