Future View: Web Navigation based on Learning User's Browsing Patterns

Norikatsu Nagino
CISS, IGSSE, Tokyo Institute of Technology
4259 Nagatsuta, Midori
Yokohama 226-8502, Japan
nagino@ntt.dis.titech.ac.jp

Seiji Yamada
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda
Tokyo 101-8430, Japan
seiji@nii.ac.jp

Abstract

In this paper, we propose a Future View *system that assists user's usual Web browsing. The* Future View *will prefetch Web pages based on user's browsing strategies and present them to a user in order to assist Web browsing. To learn user's browsing patterns, the* Future View *uses two types of learning classifier systems: a content-based classifier system for contents change patterns and an action-based classifier system for user's action patterns. The results of learning is applied to crawling by Web robot, and gathered Web pages are presented to a user through a Web browser. We experimentally show effectiveness of navigation using the* Future View*.*

1 Introduction

The World Wide Web is available to gather interesting information for a user. There are Web pages over almost all fileds. However, finding objective Web pages is very hard for a user because of the width of the Web, therefore empirical browsing strategies are very important for user's efficient browsing. Many techniques to assist users on their browsing tasks have been developed. For example, there are many methods of gathering relative Web pages about some keywords[2], and recommending next links or relative Web pages for a user from the current Web page[4]. Information of Web page contents is mainly used in those techniques. However, they are not enough for crawlers to narrow their search spaces. There are also some techniques to assist users on their browsing tasks by learning accessed Web pages sequences[6, 7]. However they assist a user in a closed space on a same Web site because they use logs on a Web server. It is difficult to assist user's Web browsing in a open space because of a wide search space. However, we can make the search space sufficiently narrow, if we consider not only similarity of Web pages but also user's strategic search patterns.

In this paper, we propose a Future View system assists user's browsing tasks. The Future View learns user's browsing patterns, and prefetches Web pages according to user's browsing strategies by applying learned browsing patterns for crawling Web pages by Web robot. The Future View uses a content-based learning technique and a user's action-based learning technique to learn user's browsing patterns. These two types of learning are developed with evolutionary learning method, e.g. classifier systems. Gathered Web pages are listed in a user interface. If the results correspond to user's interest and include objective Web pages, he/she can access to the Web pages directly through a user interface. On the other hand, if the results do not correspond to interest of a user, he/she must change his/her browsing strategy.

2 Future View Architecture

Figure 1 shows an overview of the Future View. The Future View consists of two main components. The first

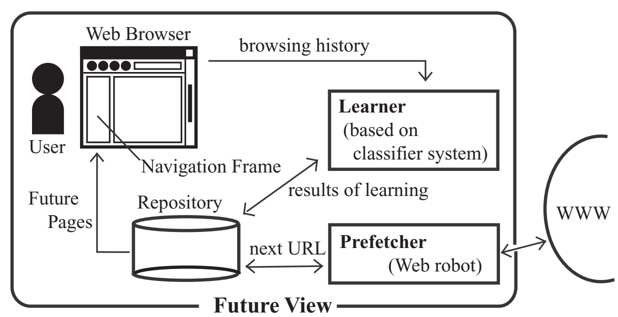

Figure 1. System overview

component Learner is a learning component based on classifier systems(CS)[3]. It receives information about accessed Web pages from an altered Web browser "Mozilla" provided as open source. The Learner is constructed with two types of classifier systems: a Content-based CS(CCS) and an Action-based CS(ACS) (Fig. 2). A CCS learns user's browsing patterns based on contents of accessed Web pages. On the other hand, a ACS learns user's browsing patterns based on user's actions. Accessed Web page information is transformed to a value whether it is included in a "Page Class". A Page Class is an abstract representation of Web pages set based on various features of its. A condition part and an action part of a classifier consist of values for Page Classes. We call page classes for a CCS "Content-based Page Class(CPC)" and page classes for an ACS "Action-based Page Class(APC)". We show the detail of its in the section 3.1 and 3.3. The second component is a Prefetcher. Our crawler prefetches Web pages referring the results of learning classifiers. A crawler stores the searching states at the point in time with information of Web pages. And in order to prefetch more deep Web pages, prefetched Web pages are considered as a part of browsing history. Gathered Web pages are always displayed as *Future pages* in the *Navigation Frame* with a function of a Web browser "Sidebar", while browsing.

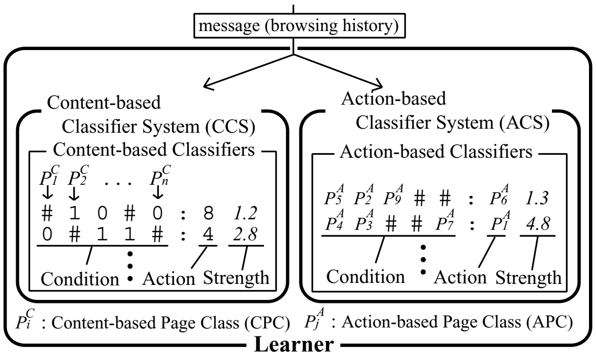

Figure 2. Architecture of a Learner

3 Learning User's Browsing Patterns

3.1 Design of CPCs

In this section, we show the way of designing CPCs. When a user browses on a topic, the topic can be characterized with appeared words in Web pages on the topic. We can define CPCs based on some viewpoints below using *Browsing Session* in order to distinguish the type of words aimed by a user. The "browsing session" means a browsing sequence on a single topic, whether a user access to related Web pages from many links pages or using search engines.

- **topic continuity**
  - Web pages which include top n words of high TFIDF values in the title or body or anchor text.
    - ∗ The TFIDF[5] values here for all words in the current document are calculated with the other documents accessed before the current session.
  - Web pages accessed by following links of which anchor text include keywords input for search engines in the current session.
- **content difference on a topic**
  - Web pages which include more words with high TFIDF value and not included the other Web pages in the same session.
    - ∗ The TFIDF values here for each word in the current session are calculated with only documents in the current session.
  - Web pages which the very similar Web pages didn't appear in the current session.

3.2 Learning with a CCS

Each classifier consists of CPCs for a CCS are represented as below.

$$\langle \text{classifier} \rangle = \langle \text{condition-part} \rangle : \langle \text{action-part} \rangle$$
$$\langle \text{condition-part} \rangle = C_1^C, \ldots, C_n^C = \{1, 0, \#\}^n$$
$$\langle \text{action-part} \rangle = A^C = \{P_1^C, \ldots, P_n^C\}$$

A C_i^C means a i-th component of a condition-part. When a CPC corresponding to a i-th component is represented as a P_i^C and the current Web page is represented as p_{curr}, each component C_i^C means: if the C_i^C equals 1, then a p_{curr} is included in a page class P_i^C, and if the C_i^C equals 0, then a p_{curr} is not included in a page class P_i^C, and if the C_i^C equals "#", then whether a p_{curr} is included in a page class P_i^C or not. Here, A^C means a CPC which the next Web page p_{next} will be included in . Whenever a user visit a new Web page, a set of values for matching is constructed with a message (browsing history) and it is compared with a condition part of a classifier for matching. A CCS performs updating strength of each classifier using the *Bucket-Brigade* algorithm in the reinforcement component and GA in the discovery component like standard classifier systems.

3.3 Design of APCs

In this section, we show the way of designing APCs. An APC represents a set of Web pages based on user's actions used in an ACS, and the classifiers consists of the APCs represent a user's characteristic browsing pattern. We provide a guideline for defining APCs and samples of APCs below.

- **interests to news**
  - Web pages accessed by following new added links.
- **strategic search**
  - Search engine's top page accessed by directly inputting the URL to a browser.
  - Web pages which accessed by following a link near by the previous followed link and were not followed today yet.

3.4 Learning with an ACS

The Future View uses an ACS to learn user's browsing patterns based on user's actions. Each classifier consist of APCs for a ACS are represented as below.

$$\langle\text{classifier}\rangle = \langle\text{condition-part}\rangle : \langle\text{action-part}\rangle$$
$$\langle\text{condition-part}\rangle = C_1^A, \ldots, C_n^A = \{P_1^A, \ldots, P_m^A, \#\}^n$$
$$\langle\text{action-part}\rangle = A^A = \{P_1^A, \ldots, P_m^A\}$$

Here, i means the order of access to a Web page, and C_i^A means a APC in which the i-th Web page should be included. The current Web page is included in an APC of C_n^A, and a Web page accessed at the previous step is included in an APC of C_{n-1}^A. When the current Web page is represented as p_{curr}, each component C_i^A means: if the C_i^A equals P_j^A, then a p_{curr} is included in a page class P_j^A, and the C_i^A equals "#", then whether a p_{curr} is included in any page class or not. Here, A^A means a APC which the next Web page p_{next} will be included in . Whenever a user visits a new Web page, a set of values for matching is constructed with a message (browsing history) and it is compared with a condition part of a classifier for matching. An ACS updates strength of each classifier based on the *Bucket-Brigade* algorithm in the reinforcement component and GA in the discovery component like standard classifier systems in the same way to an CCS. In addition, we use the following techniques for learning efficiency.

- *Partial Matching*[1]: The Future View performs partial matching sequence instead of exact matching.
- Using page classes and *instances* for classifiers: The Future View uses not only page classes but also URLs of Web pages as instances.
- *Cover (detector) operator*[8]: In a discovery component of an ACS, new classifiers are created a matching classifiers out of the current message when there is no classifiers match the current message.

4 Prefetching and Presenting Future Pages

A prefetcher gathers Web pages as *Future Pages* start with the user's current Web page applying learned browsing patterns, and stores them to a repository. A prefetcher search *Future Pages* with the standard *best-first search* algorithm with a evaluation function using values of the last applied classifier's strength.

Future Pages are presented to a user through a browser. Fig.3 show a user interface. A left upper frame display URLs as a browsing history and the "New Session" button to split browsing sessions. A left lower frame indicates a list of *Future Pages* with a title, a URL string, a thumbnail image of the Web page, a button to display a trail from the current Web page and a part of contents of the Web page. A user can access the Web page by clicking a URL link.

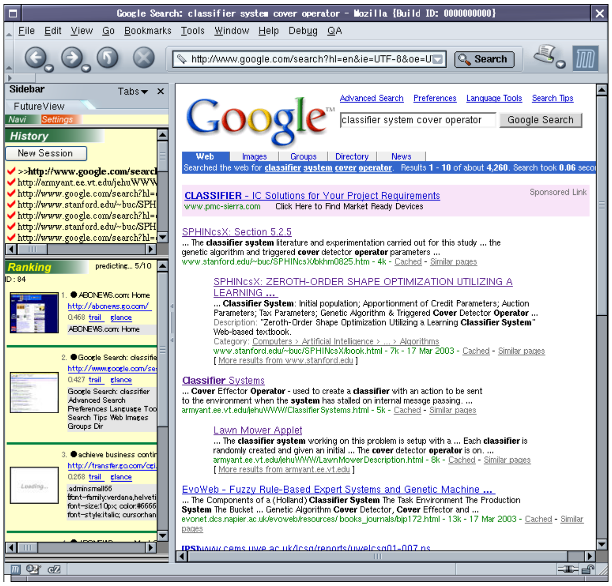

Figure 3. User Interface

5 Experiments

In order to investigate effect of the Future View, we evaluate the results of using the Future view by a few members of an our laboratory. We show *Future Pages* which are presented by the Future View and the results of evaluating qualities of presented *Future Pages* by a user.

5.1 Example of Future Pages

We show an interesting pattern in this section. A user has performed the following browsing occasionally.

- He has used a search engine sometimes.
- After he has inputed words to a search engine, he has selected some results from the search engine one by one from the top.
- After he has selected some results from the search engine, he has has accessed to a News site which have been often used.
- He has selected some new articles.

A user visited a Web page about "Linux OS", and he broke off the current session by pushing "New Session" button, and he accessed to the top page of search engine "google". Now he have just selected some results from search engine and back to the results pages of search engine (P_2 in Fig. 4). Then, Web robot visited Web pages P_j, P_2, P_k, P_l, P_k(Fig. 4), and Web Pages P_k, P_2, P_l, P_j are presented as *Future Pages*.

5.2 Qualities of Future Pages

We investigated qualities of *Future Pages* in order to confirm effectiveness compared with the other systems; there

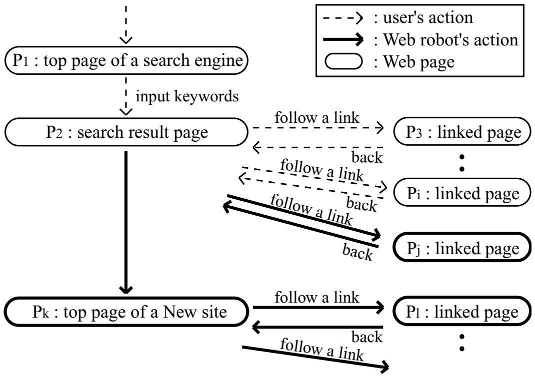

Figure 4. Example of interesting pattern

are not appropriately comparable systems considered the open search space, however. Therefore we investigate the difference of effects between an ACS and a CCS. In order to investigate the different effects between an ACS and a CCS, two subjects evaluated up to 10 *Future Pages*(if there is a few Web pages for presenting the user, the number of *Future Pages* may less than 10 pages) for 5 current Web pages at each case that a prefetcher is applied the results of learning of ACS and CCS, only ACS and only CCS for the comparison(150 or less *Future Pages* for a subject in total). Each *Future Pages* is evaluated with 0–4 points: 4 for very interesting Web pages, 0 for indifferent Web pages and others for intermediate values. The Table1 shows average values of evaluations. We also compare the results for two types of user's browsing, with or without the specific goal. Each user evaluates *Future Pages* which are gathered when he/she was browsing to gather technical papers(w_1) and when he/she was browsing without any goal(w_2).

	subject A		subject B	
	w_1	w_2	w_1	w_2
$e_{c,a}$	**1.45** (42)	**2.06** (50)	**1.47** (37)	1.28 (36)
e_c	**1.41** (49)	0.40 (26)	**1.09** (42)	1.14 (32)
e_a	1.40 (38)	**1.87** (43)	0.89 (48)	**1.82 (16)**

$n(m)$: n is an average values of evaluations and m is the number of presented *Future Pages* (max. is 50).
$e_{c,a}$: gathering *Future Pages* with both a CCS and an ACS.
e_c: gathering *Future Pages* with an only CCS.
e_a: gathering *Future Pages* with an only ACS.

Table 1. Evaluation of Future Pages

For both subjects, the evaluation of *Future Pages* gathered with an only CCS were better than with an only ACS at w_1. On the other hand, the evaluation of *Future Pages* gathered with only an ACS were better than with only an CCS at w_2. This means that a CCS outperforms an ACS in searching specific information, and an ACS outperforms a CCS for user's arbitrary browsing. Moreover, the evaluation of *Future Pages* gathered with both a CCS and an ACS were almost better than others. However, the evaluation of *Future Pages* gathered with both a CCS and an ACS were worse than with an only ACS for the subject B at w_2. We consider that the reason is the difference of the number of presented *Future Pages* between e_c and e_a. The number of presented *Future Pages* with an only ACS for a subject B were actually a little at w_2. And presented *Future Pages* of $e_{c,a}$ for the subject B at w_2 include many ones of e_c than e_a. Therefore the evaluation value of e_c lead the evaluation value of $e_{c,a}$ to a low value.

The results of this experiments shows that a CCS and an ACS can work suitably for some kind of user's browsing. We confirmed that the Future View seemed to all users to working effectually for all browsing, and it worked seamlessly even when he/she change his/her browsing strategies according to the situation.

6 Conclusions

We proposed a Future View system that assists user's Web browsing on the Web as an open space. We also show a concept of "*Future Pages*" gathered with user's browsing patterns. The Future View gathers relative Web pages to the current Web pages or keywords. A CCS and an ACS are components for learning user's browsing patterns. We provide policies for designing them. We also investigated learned browsing patterns and its results, and verified the effectiveness of presented *Future Pages* by a Future View.

References

[1] L. B. Booker. Improving the Performance of Genetic Algorithms in Classifier Systems. In *Proc. of the International Conference on Genetic Algorithms and Their Applications*, pages 80–92, Pittsburgh, PA, 1985.

[2] P. D. Bra and R. Post. Information Retrieval in the World-Wide Web: Making Client-based Searching Feasible. *Computer Networks and ISDN Systems*, 27(2):183–192, 1994.

[3] J. H. Holland and J. S. Reitman. Cognitive Systems Based on Adaptive Algorithms. In *Pattern-Directed Inference Systems*, pages 313–329. Academic Press, 1978.

[4] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 770–775, 1997.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[6] M. Spiliopoulou. Web usage mining for Web site evaluation. *Communications of the ACM*, 43(8):127–134, Aug. 2000.

[7] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan. Web usage mining. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM*, 1, 2000.

[8] S. W. Wilson. Knowledge growth in an artificial animal. *Proceedings Genetic Algorithms and their Applications*, pages 16–23, 1985.