Browsing Support by Hilighting Keywords based on a User's Browsing History

Yutaka Matsuo

National Institute of
Advanced Industrial Science and Technology
Aomi 2-41-6, Tokyo 135-0065, JAPAN
y.matsuo@aist.go.jp

Hayato Fukuta Mitsuru Ishizuka

Graduate School of Information Science and Technology,
University of Tokyo
Hongo 7-3-1, Tokyo 113-8656, JAPAN
{hfukuta,ishizuka}@miv.t.u-tokyo.ac.jp

Abstract— **We develop a browsing support system which learns user's interests and highlights keywords based on a user's browsing history. Monitoring the user's access to the Web enable us to detect "familiar words" for the user. We extract keywords, which are relevant to the familiar words in the current page, and highlight them. The relevancy is measured by the biases of co-occurrence, called *IRM* (Interest Relevance Measure). Our system consists of three components; a proxy server which monitors access to the Web, a frequency server which stores frequency of words in the accessed Web pages, and a keyword extraction module. Preliminary reports are shown to evaluate the system.**

Keywords— **browsing support, user profile, keyword extraction, proxy server.**

I. Introduction

As the WWW matures, more and more popular sites emerge, which are designed carefully and updated frequently to attract many people. On the other hand, there are a large number of attempts to personalize the Web. Learning user preferences enables to discover Web information sources that correspond to these preferences, and possibly those of other individuals with similar interests [2]. This paper also tries to deal with the problem of "how to personalize the Web."

In this paper, we develop a personalized browsing support system which highlights keywords for a user. Our definition of "keywords" is different from the usual usage of keywords; keywords usually mean important words to represent the content of the document(s), and/or to distinguish the document from others [3]. In our definition, keywords mean important words in a document for a user: One might read a document and think some words as important, while another might read the same document and think other words as important. Important words for a reader depend on reader's interests and context.

Let us take an newspaper article "Suzuki hitting streak ends at 23 games" (May 19, 2001) for example. Ichiro Suzuki is a Japanese Major League Baseball player who got MVP last year. One who is greatly interested in Major League Baseball will be interested in the phrase such as "hitting streak ends," because he/she knew that Suzuki was achieving the longest hitting streak in the majors in the year. On the other hand, one who doesn't have an interest on MLB at all sees the words "game" or "Seattle Mariners" as informative words, because he/she can get that this article is written about baseball, and that's enough.

Therefore, relevancy between one's interest and words is good criteria of keywords. If a user is not familiar with the topic, he/she may think general words of the topic as important, On the other hand, if a user is familiar with the topic, the general words are not so informative. He/she may think more detailed words as important.

In our system, we monitor the contents of accessed Web pages and count the number of occurrence of each word. Frequently appearing words are considered to be "familiar" to the user. These familiar words represent user's interests, but familiar words themselves are not interesting for the user. A word is interesting to the user if it is relevant to the familiar words but not a familiar word itself. Relevancy to the familiar words is measured by our new weighting scheme, called *IRM* (Interest Relevance Measure). We help a user browse the Web by highlighting the keywords. Users can grab the overview quickly and find possibly interesting words at once.

The rest of the paper is organized as follows. In the following section, we first explain *IRM*. Then the system architecture is shown in Section 3 and the experiment for evaluation is shown in Section 4. We discuss our approach in Section 5 and finally conclude this paper.

II. How to Detect Keywords?

A Web page has textual content, except for non-text pages which include images, audio, video and so on. For each page with textual content, we can get a set of words. Below we use "textual content of a page" and "a document" interchangeably.

In the context of information retrieval, words are weighted by various measures. The simplest weighing measure is based on the occurrence of words in a document. The weight of word w_i is defined as $I_{ij} = f(w_{ij})$, where $f(w_{ij})$ is frequency of w_i in document j. Another measure is based on the occurrence of a word in a document relative to its occurrence in the other documents in the database. A popular measure, called $tf \cdot idf$, is defined by

$$I_{ij} = f(w_{ij}) \cdot \log_2 \frac{n(D)}{\sum_j g(w_{ij})},$$

where $n(D)$ is the number of documents, and $g(w_{ij})$ gives 1 to word i in document j.

TABLE I
FREQUENCY AND PROBABILITY DISTRIBUTION.

Frequent term	a	b	c	d	e	f	g	h	i	j	Total
Frequency	203	63	44	44	39	36	35	33	30	28	555
Probability	0.366	0.114	0.079	0.079	0.070	0.065	0.063	0.059	0.054	0.050	1.0

a: *machine*, b: *computer*, c: *question*, d: *digital*, e: *answer*, f: *game*, g: *argument*, h: *make*, i: *state*, j: *number*

TABLE II
A CO-OCCURRENCE MATRIX.

	a	b	c	d	e	f	g	h	i	j	Total
a	—	30	26	19	18	12	12	17	22	9	165
b	30	—	5	50	6	11	1	3	2	3	111
c	26	5	—	4	23	7	0	2	0	0	67
d	19	50	4	—	3	7	1	1	0	4	89
e	18	6	23	3	—	7	1	2	1	0	61
f	12	11	7	7	7	—	2	4	0	0	50
g	12	1	0	1	1	2	—	5	1	0	23
h	17	3	2	1	2	4	5	—	0	0	34
i	22	2	0	0	1	0	1	0	—	7	33
j	9	3	0	4	0	0	0	0	7	—	23
...	...	...	...	...	...	...	...	...	...	...	...
u	6	5	5	3	3	18	2	2	1	0	45
v	13	40	4	35	3	6	1	0	0	2	104
w	11	2	2	1	1	0	1	4	0	0	22
x	17	3	2	1	2	4	5	0	0	0	34

u: *imitation*, v: *digital computer*, w:*kind*, x:*make*

The *IRM*, which we propose, is also a weighting scheme of words. This measure approximates the relevance of a word in a document and frequently appeared words in the database by counting the co-occurrence frequency.

A. Interest Relevance Measure

IRM is based on a keyword extraction method developed for technical papers [5]. We first introduce the method.

A document consists of sentences. In this paper, two terms[1] in a sentence are considered to co-occur once. That is, we see each sentence as a "basket" and we ignore term order and grammatical information except to extract word sequences.

By counting term frequencies, we can obtain frequent terms. Let us take a very famous paper by Alan Turing [9] as an example. Table I shows the top ten frequent terms (denoted as G) and the probability of occurrence, normalized so that the sum is to be 1.

Next, a co-occurrence matrix is obtained by counting frequencies of pairwise term co-occurrence, as shown in Table II. For example, term a and term b co-occur in 30 sentences in the document. Let N denote the number of different terms in the document. Because the term co-occurrence matrix is an $N \times N$ symmetric matrix, Table II shows only a part of the whole – an $N \times 10$ matrix. We do not define diagonal components here.

Assuming that term w_i appears independently from frequent terms G, the distribution of co-occurrence of term w_i and the frequent terms is similar to the unconditional distribution of occurrence of the frequent terms, shown in Table I. Conversely, if term w_i has semantic relation with a particular set of terms $g \in G$, co-occurrence of term w_i and g is greater than expected; the probability distribution is to be biased.

Figures 1 and 2 show co-occurrence probability distribution of some terms and the frequent terms. In the figures, unconditional distribution of the frequent terms is shown as "unconditional". A general term such as 'kind" or "make" is used relatively impartially with each frequent term, while a term such as "imitation" or "digital computer" shows the co-occurrence especially with particular terms. These biases are derived from either semantic, lexical, or other kinds of relation of two terms. Thus, a term with co-occurrence biases may have an important meaning in a document. In this example, "imitation" and "digital computer" are important terms as we all know: In this paper Turing proposed an "imitation game" to replace the question "Can machines think?"

Therefore, the degree of biases of co-occurrence can be used as a surrogate of term importance. However, if term frequency is small, the degree of the biases is not reliable. For example, assume term w_1 appears only once and co-occurs only with term a once (with probability 1.0). On the other extreme, assume term w_2 appears 100 times and co-occurs only with term a 100 times (with probability 1.0). Intuitively, w_2 seems more reliably biased. In order to evaluate statistical significance of biases, we use the χ^2 test, which is very com-

[1] A term is a word or a word sequence.

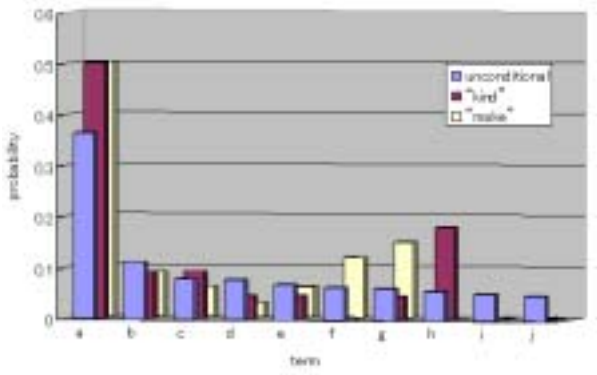

Fig. 1. Co-occurrence probability distribution of the terms *"kind"*, *"make"*, and frequent terms.

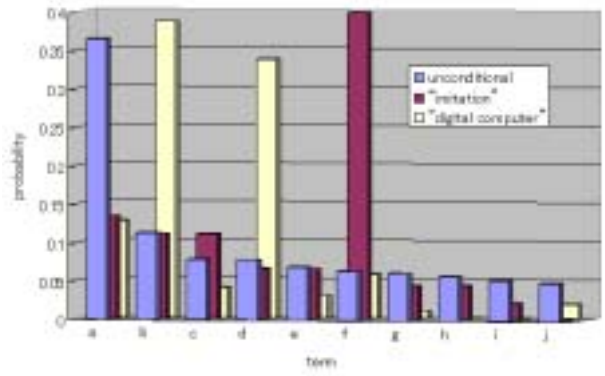

Fig. 2. Co-occurrence probability distribution of the terms *"imitation"*, *"digital computer"*, and frequent terms.

mon for evaluating biases between expected frequencies and observed frequencies. For each term, frequency of co-occurrence with the frequent terms is regarded as a sample value; a null hypothesis is that "occurrence of frequent terms G are independent from the occurrence of term w_i," which we expect to reject.

We denote the unconditional probability of a frequent term $g \in G$ as the expected probability p_g, and the total number of co-occurrence of term w_i and frequent terms G as $f_G(w_i)$. Frequency of co-occurrence of term w_i and term $g \in G$ is written as $freq(w_i, g)$. The statistics value of χ^2 is defined as follows. (We annotate a subscript to represent "in document j".)

$$\chi^2_{ij} = \sum_{g \in G} \frac{(freq(w_{ij}, g) - f_G(w_{ij})p_g)^2}{f_G(w_{ij})p_g} \qquad (1)$$

If $\chi^2(w) > \chi^2_\alpha$, the null hypothesis is rejected with significance level α (χ^2_α is normally obtained from statistical tables, or by integral calculation). The term $f_G(w_{ij})p_g$ represents the expected frequency of co-occurrence, and $(freq(w, g) - f_G(w_{ij})p_g)$ represents the difference between expected and observed frequencies. Therefore, large χ^2_{ij} indicates that co-occurrence of term w_i shows strong bias. In this paper, we use the χ^2-measure as an index of biases, not for tests of hypotheses.

TABLE III
TERMS WITH HIGH χ^2 VALUE.

Rank	χ^2	Term	Frequency
1	593.7	digital computer	31
2	179.3	imitation game	16
3	163.1	future	4
4	161.3	question	44
5	152.8	internal	3
6	143.5	answer	39
7	142.8	input signal	3
8	137.7	moment	2
9	130.7	play	8
10	123.0	output	15
⋮	⋮	⋮	⋮
551	1.0	slowness	2
552	1.0	unemotional channel	2
553	0.8	Mr.	2
554	0.8	sympathetic	2
555	0.7	leg	2
556	0.7	chess	2
557	0.6	Pickwick	2
558	0.6	scan	2
559	0.3	worse	2
560	0.1	eye	2

(We set the top ten frequent terms as G.)

Table III shows terms with high χ^2 values and ones with low χ^2 values in the Turing's paper. Generally, terms with large χ^2 are relatively important in the document; terms with small χ^2 are relatively trivial.

B. Application to User's Browsing History

The above method is useful for extracting important words from a document (especially from a technical paper; they are well written). However importance of words depend not only on the document itself but also on a reader. One who is not familiar with the topic might think some general words as important, while other who is an expert of the topic might think more detailed words as important.

Therefore, we focus on "familiar words" to the user, instead of "frequent words" in the document. Familiar words are the words which a user has frequently seen in the past. As discussed below, they can be obtained by monitoring the user's browsing behavior using a proxy server. Frequency of co-occurrence with the familiar words is measured for each word, and the bias is calculated in order to extract keywords of the document for a user. The bias shows the selective relevance to the familiar words; If a word co-occurs selectively with some familiar words, it is of importance for the user. While if a word doesn't co-occur with or co-occurs impartially with the familiar words, it is of no importance for the user.

TABLE IV
FAMILIAR WORDS FOR USER 1.

Rank	Word	Frequency
1	page	34
2	Nifty	24
3	document	20
4	link	20
5	write	19
6	information	19
...	...	...
23	piano	14
24	research	13
24	think	13
...	...	...
31	music	11
31	corner	11
...	...	...
59	listen	9
59	hobby	9
59	easy	9
59	Gifu	9

TABLE V
FAMILIAR WORDS FOR USER 2.

Rank	Word	Frequency
1	game	26
2	page	25
3	found	24
4	information	24
5	server	22
6	relevant	22
...	...	...
15	update	17
15	BBS	17
15	image	17
...	...	...
57	net	9
57	copy	9
57	series	9
57	soft	9

Definition 1 Interest Relevancy Measure*(IRM) is defined as follows. For word w_i in page j for user k,*

$$IRM_{ijk} = \sum_{h \in H_k} \frac{(freq(w_{ij}, h) - f_G(w_{ij})p_h)^2}{f_G(w_{ij})p_h}, \quad (2)$$

where H_k is a set of familiar words for user k, $freq(w_{ij}, h)$ is frequency of co-occurrence of word w_{ij} and h in page j, $f_G(w_{ij})$ is the total number of occurrence of word w_{ij} in page j, and p_h is the expected probability of word h to appear.

If the value of *IRM* is large, word w_{ij} is relevant to user's familiar words. The word is relevant to the user's interests, so it is a keyword for the user. Conversely, if the value of *IRM* is small, word w_{ij} is not specifically relevant to any of the familiar words.

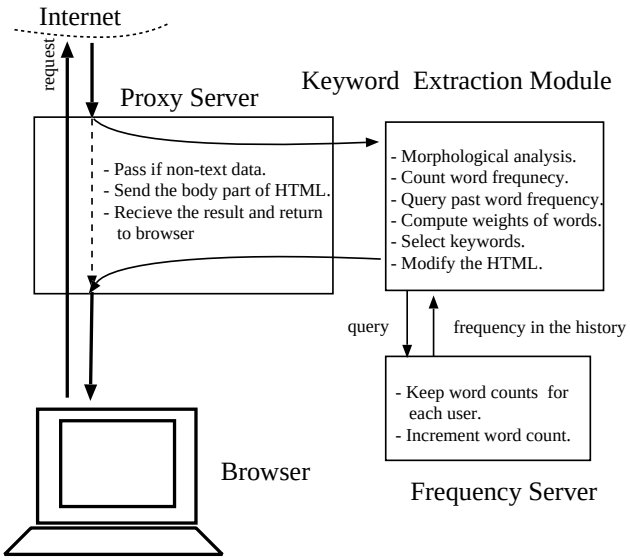

Fig. 3. System architecture.

III. System Architecture

In our system, Web pages accessed by a user are monitored by a proxy server. Then the count of each word is stored in a database. For example, after an hour browsing we can get a word list for each user shown in Table IV (by a user whose hobby is playing piano) and Table V (by a user whose hobby is game-playing). After eliminating stop words, the list represents the user's interests.

The system consists of three components; a proxy server, a frequency server, and a keyword extraction module as shown in Figure 3.

A. Proxy Server

Proxy Server inspects the browser's HTTP requests. When the response is returned, it judges whether the page is html/text or not. If it is a non-text file, or the length of the text is too short, it forwards the page to the browser without any change. Otherwise, it sends the body part of the page to Keyword Extraction Module. Then it receives the modified content where keywords are highlighted, and forward it to the browser. As the proxy creates new threads to handle the browser's requests, it allows multithreaded browsers to be able to have multiple requests pending at one time.

B. Keyword Extraction Module

Keyword Extraction Module first does morphological analysis, and count word frequency in the page. Then, it queries to Frequency Server in order to get word frequency of the past for the user. Based on the word counts and the past word counts, *IRM* of words are calculated. Selected number of words are highlighted as keywords by bold red big characters, by inserting `<b><font size=+1 color=red>` and `</b></font>` tags.

C. Frequency Server

Frequency Server keeps the total number of browsed pages and count of each word for each user. In other words, it manages user profiles. Particular words are defined as stop word; It includes stop list by Salton [7], and common words in Web pages, e.g. "copyright," "page," "link," "news," "search," "mail," and so on.

Using this system, a user can browse the Web as usual. The difference is that some words are highlighted red. Users can grab the overview quickly and find possibly interesting words at once.

IV. Evaluation

For evaluation, ten people tried this system for more than one hour. We asked them to evaluate the system. Three methods are implemented for comparison with the same stop list: The weight of word is calculated by (I) word frequency, (II) $tf \cdot idf$ measure, and (III) *IRM* measure. System (I) highlights simply the most frequent words in the document in red color, and the most familiar words in blue color. System (II) highlights the words with highest $tf \cdot idf$ value in red color, and the most familiar words in blue color. In our case, $tf \cdot idf$ value is calculated using the past frequency of word w_i for user k, $f_{past}(w_{ik})$, and the number of browsed pages n_k,

$$tfidf_{ijk} = f(w_{ij}) \cdot \log_2 \frac{n_k}{f_{past}(w_{ik})}.$$

System (III) highlights the words with highest *IRM* value in red color, and the most familiar words in blue color. The weighting algorithm of the system is kept blind to the participants. Note that in all three systems, the words in blue color are extracted in the same way.

After using each system, we ask the following questions. Answers to the questions were made on a 5-point Likert-scale from 1 (not at all) to 5 (very much).

Q1 Do this system help you browse the Web?
Q2 Are the red color words interesting to you?
Q3 Are the interesting words colored red?
Q4 Are the blue color words interesting to you?
Q5 Are the interesting words colored blue?

After evaluating all three systems, we ask the following two questions.

Q6 Which one helps your browsing the most?
Q7 Which one detects your interests the most?

The results are shown in Table VI and VII. As for the system support (Q1), the difference is small. Tfidf and *IRM* are comparable. The questions about red color words (Q2 and Q3) make differences. Though tfidf performs as well as *IRM* does with respect to precision, it performs worse with respect to recall. Q4 and Q5 are about blue color words, which are extracted similarly in the systems. Nevertheless, the evaluation of *IRM* is worse than others. (Hopefully, this is because the red color words are better.) Overall, tfidf and *IRM* performs well. But in terms of catching user's interest correctly, *IRM* performs the best.

Q6 and Q7 are more straightforward questions. Obviously, word frequency is the least useful. Although a couple of participants voted for tfidf, the most agreed *IRM* can detect words of the user's interest the most.

TABLE VI
Average point of participants.

	Q1	Q2	Q3	Q4	Q5
(I) Word frequency	2.8	3.2	2.9	2.7	2.7
(II) tfidf	3.2	4.0	3.3	2.5	2.5
(III) *IRM*	3.2	4.1	3.8	2.0	2.4

TABLE VII
Cast ballots.

	Q6	Q7
(I) Word frequency	1	0
(II) tfidf	3	2
(III) *IRM*	6	8

No one complained about the processing time, because the average processing time is less than a second. However, some say that changing fonts of HTML sometimes destroys the design of the page.

V. Discussion and Related Work

Although *IRM* is a different form of tfidf, they have some qualitative properties in common.

- If a word appears little in the document, the weight is small: Because *IRM* measures the significance of biases, a small number of appearance of words often implies small significance.
- If a word is familiar to the user (i.e. frequently appeared in the past), the weight is small.

However, the main difference is the following.

- Even if a word appears frequently in the document, the weight of the word is small if it is not relevant to user's interests (i.e. if it doesn't co-occurs with the familiar words).

Although our system is currently for Japanese language, we have shown the merit of our system in previous section.

In recent years, various systems have been developed which utilize user models for personalization: Letizia [4] learns the interests of a user by observing their browsing behavior. Then it recommends links to follow. WebACE [2] proposes an agent for exploring and categorizing documents on the WWW. It uses tfidf measure for feature vector of the documents, and clusters these documents. Somlo presents an agent which maintains a history list with addresses of all the sites visited by a user [8]. If repetition occurs, the agent will learn this and add the address to the user profile. The profile categories are based on tfidf measure. Web Personae is the personalized search and browsing system [6]. It models users with multiple profiles, each corresponding to a distinct topic or domain. WebMate [1] is an agent that assists browsing and searching. It represents different domains of user interest using multiple term vectors.

These researches basically use word frequency or tfidf measure. However, tfidf measure lacks the consideration of relevance to user's interests. Our *IRM* measure may

contribute to weight words based on both frequency in the documents and user's interests.

Though each individual user has a number of interests not necessarily related to each other, our system can properly handle the interests; If a word co-occurs selectively with some familiar words, it is highlighted. Other familiar words have little effect on the bias.

VI. Conclusions

In this paper, we proposed a new word weighting scheme, called *IRM* to measure the relevance of a word and user's interests. Then we developed a browsing support system using a proxy server, which detects user's interests by monitoring the access to the Web. We have shown some preliminary result that our system highlights interesting words for a user.

References

[1] L. Chen and K. Sycara. WebMate: A personal agent for browsing and searching. In *Proceedings 2nd International Conference on Autonomous Agents*, 1998.

[2] E. Han, D. Boley, M. Gini, R. Gross, and K. Hastings. WebACE: A web agent for document categorization and exploration. In *Proceedings 2nd International Conference on Autonomous Agents (Agents'98)*, 1998.

[3] K. Kageura and B. Umino. Methods of automatic term recognition. *Terminology*, 3(2):259–289, 1996.

[4] H. Lieberman. Letizia: An agent that assists Web browsing. In *Proceedings 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, 1995.

[5] Y. Matsuo and M. Ishizuka. Keyword extraction from a document using word co-occurrence statistical information. *Transactions of the Japanese Society for Artificial Intelligence*, 17(3), 2002.

[6] JP McGowan, N. Kushmetrick, and B. Smyth. Who do you want to be today? Web Personae for personalised information access. In *Proceedings International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2002.

[7] G. Salton. *Automatic Text Processing*. Addison-Wesley, Mass., 1989.

[8] G. L. Somlo and A. E. Howe. Agent-assisted internet browsing. In *Workshop on Intelligent Information Systems (AAAI-99)*, 1999.

[9] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433, 1950.