Word Weighting based on User's Browsing History

Yutaka Matsuo[1]

National Institute of Advance Industrial Science and Technology,
y.matsuo@aist.go.jp,
WWW home page: http://www.carc.aist.go.jp/~y.matsuo

Abstract. We developed a word-weighting algorithm based on the information access history of a user. The information access history of a user is represented as a set of words, and is considered to be a user model. We weight words in a document according to their relevancy to the user model. The relevancy is measured by the biases of co-occurrence, called *IRM* (Interest Relevance Measure), between a word in a document and words in the user model. We evaluate *IRM* through a constructed browsing support system, which monitors word occurrences on the user's browsed Web pages and highlights keywords in the current page. Our system consists of three components: a proxy server that monitors access to the Web, a frequency server that stores the frequencies of words appearing on the accessed Web pages, and a keyword extraction module.

1 Introduction

Currently, many information support systems combined with natural language techniques use *tfidf* to measure the weight of words. *Tfidf*, based on statistics of word occurrence on a target document and a corpus, has been shown to be effective in many practical systems including summarization systems and retrieval systems [7]. Its effectiveness is also supported from an information theoretical view [1].

However, a word that is important to one user is sometimes not important to others. Let us take the newspaper article "Suzuki hitting streak ends at 23 games", for example. Ichiro Suzuki is a Japanese Major League Baseball player who was recognized as MVP in 2001. A user who is greatly interested in Major League Baseball would be interested in the phrase such as "hitting streak ends," because he/she would know that Suzuki was achieving the longest hitting streak in the majors in that year. On the other hand, a user who has no interest in MLB at all would note the words "game" or "Seattle Mariners" as the informative words, because those words would indicate that the subject of the article was baseball, and that knowledge would be sufficient.

Current systems utilize the weight of words to represent a user profile, and to compare a document profile with a user profile. However, word weighting and keyword selection that reflect a user's interest are important because appropriate selection of keywords improves the accurcy of the comparison between a document profile and a user profile.

Table 1. Frequency and probability distribution.

Frequent word	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*

In order to measure the weight of words more correctly, contextual information about a user is necessary. This paper shows one approach to address context-based word weighting, focusing on the statistical feature of word occurrence: If a user is not familiar with the topic, he/she may think general words related to the topic are important. On the other hand, if a user is familiar with the topic, he/she may think more detailed words are important.

The rest of the paper is organized as follows. In the next section, we explain *IRM*. We evaluate IRM by constructing the browsing support system shown in Sections 3 and 4. We discuss our approach in Section 5 and offer concluding remarks.

2 Weighting Words

2.1 Weighting by Co-occurrence Biases

IRM is based on a word-weighting algorithm applied to a single document. We first introduce the method [5].

A document consists of sentences. Here, two words[1] in the same sentence are considered to co-occur once. By counting the word frequencies, we can obtain frequent words. Let us take a very famous paper by Alan Turing [10] as an example. Table 1 shows the top ten frequent words (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 word co-occurrence, as shown in Table 2. For example, word a and word b co-occur in 30 sentences in the document. Let N denote the number of different words in the document. Because the word co-occurrence matrix is an $N \times N$ symmetric matrix, Table 2 shows only a part of the whole – an $N \times 10$ matrix. We do not define diagonal components here.

Assuming that word w_i appears independently from frequent words G, the distribution of co-occurrence of word w_i and any of the frequent words is similar to the unconditional distribution of occurrence of the frequent words, which is shown in Table 1. Conversely, if word w_i has a semantic relation with a particular set of words $g \in G$, co-occurrence of word w_i and g is greater than expected; the probability distribution is biased.

[1] In this paper, we refer to a word as a word or a word sequence.

Table 2. 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*

Looking at Table 2, a general word such as 'kind" or "make" is used relatively impartially with each frequent word, while a word such as "imitation" or "digital computer" shows co-occurrence especially with particular words. These biases are derived from either semantic, lexical, or other kinds of relation between two words.

Therefore, the degree of biases of co-occurrence can be used as a surrogate for word importance. In order to evaluate the statistical significance of biases, we use the χ^2 test. We denote the unconditional probability of a frequent word $g \in G$ as the expected probability p_g, and the total number of co-occurrences of word w_i and any of the frequent words G as $f_G(w_i)$. The frequency of co-occurrences of word w_i and word $g \in G$ is written as $freq(w_i, g)$. The statistical value of χ^2 is defined as follows.

$$\chi_i^2 = \sum_{g \in G} \frac{(freq(w_i, g) - f_G(w_i)p_g)^2}{f_G(w_i)p_g} \tag{1}$$

The word $f_G(w_i)p_g$ represents the expected frequency of co-occurrence, and $(freq(w, g) - f_G(w_i)p_g)$ represents the difference between expected and observed frequencies. Therefore, large χ_i^2 indicates that co-occurrence of word w_i shows a strong bias.

Table 3 shows words with high χ^2 values in Turing's paper. Generally, words with large χ^2 are relatively important in the document; words with small χ^2 are relatively trivial. This method performs better than *tf*, and comparably to *tfidf* [5].

Table 3. Words with high χ^2 value.

χ^2 value	frequency	label
196.9	16	imitation game
88.9	15	play
62.4	9	human computer
60.1	3	card
57.1	4	future
50.4	10	logic
45.1	7	identification
44.4	6	universality
42.7	30	state

2.2 Interest Relevance Measure

In the above method, the selection of frequent words G is essential to the resultant weight, because for each word, the co-occurrence with $g \in G$ is counted and contributes to the χ^2 value. If we set G differently, the obtained weighting will also become different.

For example, if we add the word "logic" to the frequent words G in Turing's paper, we get the result shown as in Table 4. "logic system" and "proposition" have high values because these words co-occur with "logic". If we add the word "God" to G, we get the result shown in Table 5. Now "animal," "woman," and "book" appear because these words co-occur selectively with "God". By adding the word w to G, words relevant to w appear important because they co-occur with w. This agrees with our intuition: for example, if a user is interested in motorbikes, he/she would likely pay attention to words related to motorbike; thus, these words would have a high weight.

Therefore, we focus on "familiar words" of the user, instead of "frequent words" in the document. Familiar words are the words which a user has frequently seen in the past. They can be obtained by, for example, monitoring the user's browsing behavior using a proxy server as discussed below. Frequency of co-occurrence with the familiar words is measured for each word, and the bias is calculated in order to measure the weight of words for a user. The bias shows the selective relevance of these words to the familiar words; if a word co-occurs selectively with several familiar words, it is of importance to the user. On the other hand, if a word does not co-occur, or co-occurs impartially, with each of the familiar words, it may not be important to the user.

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

$$IRM_{ik} = \sum_{h \in H_k} \frac{(freq(w_i, h) - f_G(w_i)p_h)^2}{f_G(w_i)p_h}, \tag{2}$$

Table 4. Words with high χ^2 value on the frequent words + "logic"

χ^2 value	frequency	label
196.6	16	imitation game
88.5	15	play
84.4	3	logic system
62.2	9	human computer
60.0	3	card
57.0	4	future
44.9	7	identification
44.2	6	proposition
43.9	5	limitation

Table 5. Words with high χ^2 value on the frequent words + "God"

χ^2 value	frequency	label
196.2	16	imitation game
113.8	6	animal
88.2	15	play
62.0	9	human computer
59.9	3	card
56.9	4	future
49.8	10	logic
44.7	7	identification
43.9	5	woman
40.8	5	book

where H_k is a set of familiar words for user k, $freq(w_i, h)$ is the frequency of co-occurrence of words w_i and h, $f_G(w_i)$ is the total number of occurrences of word w_i, and p_h is the expected probability of word h to appear.

If the value of *IRM* is large, word w_i is relevant to the 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_i is not specifically relevant to any of the user's familiar words.

3 Evaluation

It is difficult to evaluate *IRM* objectively because the weight of words depends on a user's familiar words, and therefore varies among users.

Therefore, we evaluate *IRM* by constructing a Web browsing support system. 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. The system, as shown in Fig. 1, consists of three components: a proxy server, a frequency server, and a keyword extraction module.

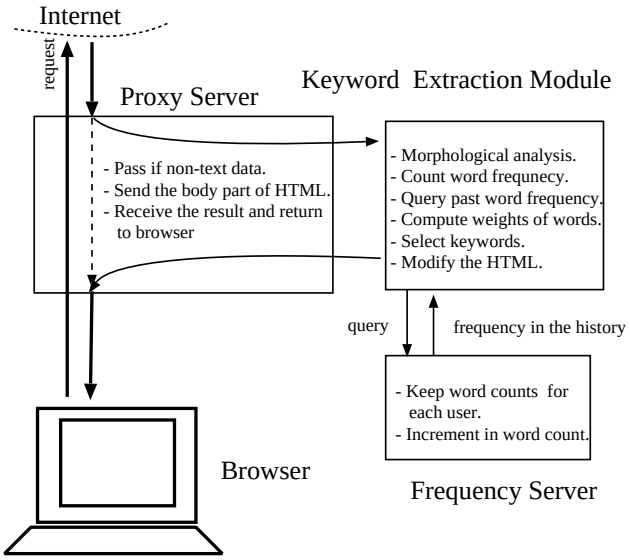

Fig. 1. System architecture.

3.1 Proxy Server

The Proxy Server inspects the browser's HTTP requests. When the response is returned, it judges whether the page is html/text. If it is a non-text file, or the length of the text is too short, it forwards the page to the browser without making any changes. Otherwise, it sends the body part of the page to the Keyword Extraction Module. Then it receives the modified contents in which the keywords are highlighted, and forwards it to the browser. Because the proxy server creates new threads to handle the browser's requests, it allows multiple pending requests from multithreaded browsers.

3.2 Keyword Extraction Module

The Keyword Extraction Module first performs morphological analysis, and counts the word frequencies on a page. Then it queries the Frequency Server in order to obtain the word frequency of the past for the user. Based on current and past word counts, the *IRMs* of various words are calculated. A given number of selected words are highlighted as keywords in a bold, red, larger font by inserting `<b><font size=+1 color=red>` and `</b></font>` tags.

3.3 Frequency Server

The Frequency Server keeps a record of the total number of browsed pages and a count of each word for each user. In other words, it manages user profiles. Particular words are defined as stop words; This includes the stop list by Salton [8],

and words common to Web pages, such as "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 as described. Users can grab the overview quickly and locate possibly interesting words at once.

4 Evaluation

For purposes of evaluation, ten people each tried this system for more than one hour. We asked them to evaluate the system. Three methods were implemented for comparison, all using the same stop list: The weight of a word was calculated by (I) word frequency, (II) $tf \cdot idf$ measure, and (III) *IRM* measure. System (I) simply highlights the most frequent words in the document in red, and the most familiar words in blue. System (II) highlights the words with highest $tf \cdot idf$ value in red, and the most familiar words in blue. In our case, the $tf \cdot idf$ 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, as follows:

$$tfidf_{ik} = f(w_i) \cdot \left(\log_2 \frac{n_k}{f_{past}(w_{ik})} + 1 \right).$$

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

After the user had tried each system, we asked him/her following questions. Answers to the questions were made on a 5-point Likert-scale from 1 (not at all) to 5 (very much).

Q1 Did this system help you browse the Web?
Q2 Are the words in red of interest to you?
Q3 Do the interesting words appear in red?
Q4 Are the words in blue of interest to you?
Q5 Do the interesting words appear in blue?

After the user had evaluated all three systems, we asked him/her the following two questions.

Q6 Which system assisted your browsing the most?
Q7 Which system best detected your interests?

The results are shown in Tables 6 and 7. With regard to the types of system support (Q1), the difference among them was small. *Tfidf* and *IRM* were comparable. The questions regarding the red-highlighted words (Q2 and Q3) demonstrated differences. Though *tfidf* performed as well as *IRM* with respect to precision, it performed worse with respect to recall. Q4 and Q5 about blue-highlighted words were similarly extracted in the three systems. Nevertheless,

Table 6. 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 7. Cast ballots.

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

IRM was evaluated as worse than the other systems. (Hopefully, this is because the words highlighted in red were better selected.) Overall, *tfidf* and *IRM* performed well. However, in terms of catching the user's interest, *IRM* performed best.

Q6 and Q7 are more straightforward questions. Obviously, word frequency is the least useful. Although a couple of participants chose *tfidf* as most effective, the majority of users agreed that *IRM* could best detect words of interest to the user.

None of the users negatively remarked the processing time, because the average processing time is less than a second. However, did remark some that changing fonts in HTML had a destructive effect on the design of the page. The performance of those three systems appeared useful for Web pages with relatively long text – for example, news articles.

5 Discussion and Related Work

Although *IRM* and *tfidf* are different algorithms, they have several qualitative properties in common.

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

The main difference of *IRM* to *tfidf* is the following.

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

This merit of *IRM* is reflected in Q2 in the previous section.

In recent years, various systems have been developed that utilize user models for personalization: Letizia [4] learns the interests of a user by observing his/her browsing behavior. Then it recommends links to follow. WebACE [3] proposes an agent for exploring and categorizing documents on the Web. It uses the *tfidf* measure for the feature vector of documents, and clusters these documents. Somlo presents an agent that maintains a history list with addresses of all the sites visited by a user [9]. If repetition occurs, the agent will learn this and add the address to the user profile. The profile categories are based on the *tfidf* measure. Web Personae, a personalized search and browsing system, models users with multiple profiles, each corresponding to a distinct topic or domain [6]. WebMate [2] is an agent that assists the user in browsing and searching. It represents different domains of user interest using multiple word vectors.

The above mentioned systems basically use word frequency or *tfidf* measure. Our *IRM* measure may contribute weight to words based both on their frequency in the documents and the user's interests.

Though each individual user may have a number of unrelated interests, our system can properly handle these; If a word co-occurs selectively with some familiar words, it is highlighted. Other familiar words have little effect on the bias.

6 Conclusions

In this paper, we proposed a new word-weighting algorithm called *IRM* for measuring the relevance of a word and a user's interests. We developed a browsing support system to evaluate *IRM*, which monitors a user's access to the Web and highlights keywords. Although the importance of a document or a sentence is not the summation of the weight of the words used in it, it is useful to calculate the weight of words in order to gauge a user's interests and consequently personalize retrieval or summarization systems.

References

1. A. Aizawa. The feature quantity: An information theoretic perspective of *thidf*-like measures. In *Proc. of SIGIR 2000*, pages 104–111, 2000.
2. L. Chen and K. Sycara. WebMate: A personal agent for browsing and searching. In *Proc. 2nd International Conference on Autonomous Agents (Agents '98)*, 1998.
3. E. Han, D. Boley, M. Gini, R. Gross, and K. Hastings. WebACE: A web agent for document categorization and exploration. In *Proc. 2nd International Conference on Autonomous Agents (Agents '98)*, 1998.
4. H. Lieberman. Letizia: An agent that assists Web browsing. In *Proc. 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. J. P. McGowan, N. Kushmetrick, and B. Smyth. Who do you want to be today? Web Personae for personalised information access. In *Proc. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2002.
7. A. Pretschner and S. Gauch. Personalization on the web. Technical Report ITTC-FY2000-TR-13591-01, The University of Kansas, 1999.
8. G. Salton. *Automatic Text Processing*. Addison-Wesley, MA., 1989.
9. G. L. Somlo and A. E. Howe. Agent-assisted internet browsing. In *Workshop on Intelligent Information Systems (AAAI-99)*, 1999.
10. A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–450, 1950.