Adaptive User Profile for Information Retrieval from the Web

Phaitoon Srinil*, and Ouen Pinngern**

** Faculty of Science and Art, Burapa University Chantaburi Campus,*
Tambon Khamong, Amphur Thamai, Chantaburi, Thailand 22170
(E-mail: bigtoon2000@yahoo.com)

***Department of Computer Engineering Faculty of Engineering,*
Research Center for Communication and Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand 10520
(E-mail: kpouen@kmitl.ac.th)

Abstract: This paper proposes the information retrieval improvement for the Web using the structure and hyperlinks of HTML documents along with user profile. The method bases on the rationale that terms appearing in different structure of documents may have different significance in identifying the documents. The method partitions the occurrence of terms in a document collection into six classes according to the tags in which particular terms occurred (such as Title, H1-H6 and Anchor). We use genetic algorithm to determine class importance values and expand user query. We also use this value in similarity computation and update user profile. Then a genetic algorithm is used again to select some terms from user profile to expand the original query. Lastly, the search engine uses the expanded query for searching and the results of the search engine are scored by similarity values between each result and the user profile. Vector space model is used and the weighting schemes of traditional information retrieval were extended to include class importance values. The tested results show that precision is up to 81.5%.

Keywords: Weight-term scheme, WWW, search engine, genetic algorithm, user profile, Information Retrieval.

1. INTRODUCTION

In this paper, we proposed a method to improve the results that retrieve from search engine. We analyzed the structure and hyperlinks of HTML documents. Then, classified terms that appear in HTML tags such as Title, H1-H6 and Anchor. Each term that appear in different location or different structure in HTML document has different importance. We gave different weight to each term by CIV (Class Importance Value) using genetic algorithm to find optimized CIV. From the result, the precision increased up to 81.5%.

Our method used vector space model [1, 2] to represent documents in weight terms of vector. Each weight terms vector was considered in two factors. First factor is termed frequency, $tf_{i,j}$, number of times term k_i appears in document d_j. Second factor, document frequency, df_i, is the number of documents d_j that has term k_i. Value of idf_i is an inverse document frequency of k_i in collection: $idf_i = log(N/df_i)$ where N is number of documents in collection. So we calculated weight-term from $w_i=tf_i*idf_i$. Each user query was also represented by weight term vector. So we calculate the value of similarity from operation of vector using cosine function.

Two differences of HTML documents and documents in Traditional IR System (TIRS) are:

1. HTML documents have structures following HTML tags. These structures determine the content in documents, while TIRS have no content structures.
2. HTML documents have links that can be analyzed the meaning of document's content. Generally, webmaster can add some descriptions in Anchor tags to explain the link, while TIRS has only terms to explain the content of documents.

2. INDEX CONSTRUCTION IN HTML DOCUMENTS

2.1 Documents Classes

HTML documents have meaning according to there structures, so we can classify terms in HTML documents. We eliminated stop word, then classified into six groups using tag structure: Title, Header, Anchor, Strong, List, and Plain text. A class consists of terms that appear in a tag as shown in Table 1.

Table 1 The six classes of term and associated HTML tags.

Class Name	**HTML tags.**
Title	TITILE
Header	H1, H2, H3, H4, H5, H6
Anchor	A
Strong	STRONG, B, EMM, I, U
List	DL, OL, UL
Plain Text	Text

The idea of classifying terms is that terms appeared in different structure have different class importance value (CIV) to documents. Terms that were being classified as Title give details of the document. Terms that were being classified as Header give details of main structure and main topics. Terms that were being classified as Anchor give details of reference documents. Terms that were being classified as Strong give details of document's significance. Terms that were being classified as List give details of overview and conclusion. Lastly, terms that were being classified as Plain Text give the whole details of the document.

2.2 Vector Representation of HTML Documents

In TIRS, documents are modeled base on vector space model [1, 2]. Documents are represented by $d = \{t \mid t \in T\}$ where T is the set of all index terms and correlation value between term and document is represented by $F(d,t): D \times T \rightarrow [0,1]$. Consequently, these two notations form a vector $\vec{d} = \sum_{t \in T} F(d,t)$ when $F(d,t) = tf_t \times idf_t$.

From Table 1, frequency of term in a class is Term Frequency Vector (TFV): *TFV=(*tf_{c1}*,* tf_{c2}*,* tf_{c3}*,* tf_{c4}*,* tf_{c5}*,* tf_{c6}*)* where tf_{c1} is the frequency of term *t* that appear in class *c1* (Title class), tf_{c2} is the frequency of term *t* that appear in class *c2* (Header class) and so on.

$$\begin{aligned} F'(d,t) &= (TFV \bullet CIV) * idf_t \\ &= TF_{dt} * idf_t \\ TF_{dt} &= \sum_{i=1..6} TF_{c\,i}(d,t) * civ_{c\,i} \\ TF_{c\,i}(d,t) &= \frac{tf_{ci}}{Maxl_{c\,i}} \end{aligned}$$

Weight term $F'(d,t)$ is calculated by using CIV, $CIV=(civ_{c1}, civ_{c2}, civ_{c3}, civ_{c4}, civ_{c5}, civ_{c6})$ where $civ_{ci} : I[1,10]$. This civ_{ci} is the importance value of term *t* in each class of document d_j, $Maxl_{ci}$ is maximum *tf* of each class c_i.

3. THE NORMAL CIV

If we assign 1 to all elements of CIV, then $F(d,t) = F'(d,t)$ so that

$$\begin{aligned} F(d,t) &= F'(d,t) \\ F(d,t) &= (TFV \bullet CIV)\, idf_t \\ F(d,t) &= ((TFV) \bullet (1,1,1,1,1,1))\, idf_t \\ F(d,t) &= (tfv_{c1} + tfv_{c2} + tfv_{c3} + tfv_{c4} + tfv_{c5} + tfv_{c6})\, idf_t \\ F(d,t) &= \sum_{i=1..6} tfv_{c\,i} \times idf_t = tf_t \times idf_t \end{aligned}$$

Here, $F(d,t) = F'(d,t)$ because we assigned CIV=(1,1,1,1,1,1). In other words, the HTML tags were removed, which means all terms have the same importance value. So we considered only TIRS.

4. THE USER PROFILE

User profile keeps user's data that are used for query expansion and optimal CIV(see 5.1), shown in Table 2. User profiles are represented by weight term vector $\vec{u} = (w_1, w_2, \ldots, w_N \mid w_i : [0,1])$ where N is number of terms in user profile, $w_i = iff_i / iff_{\max}$ and *iff* is the influence factor of term k_i in user profile.

$$iff_i = iff_i + (c/10)tf_D \qquad (1)$$

where tf_D is frequency of term k_i appeared in recommended URL, iff_{max} is maximum *iff* of user profile and *c* is user feedback parameter as shown in Table 3. When we calculated w_i, *idf* was not used because each term appeared in user profile was a keyword that was given by user for specific concern. This term was different from the other k_i in document.

Table 2 The terms in user profile.

Terms	***iff***	***W***
Intelligent	50	0.625
Learning	20	0.250
Neuron		375
Algorithm		750
…	…	…

Table 3 The data feedback from user.

User's feedback	**C**
Very interesting	2
Interesting	1
Indifferent	0
Irrelevant	-1
Very irrelevant	-2

For user profile, we initialized *iff=1* and CIV=(1,1,1,1,1) called Normal CIV. Then, genetic algorithm finds optimal CIV(see 5.1). When user give recommended URL as a feedback, *iff* is updated following the equation (1) as shown above.

5. EXPERIMENTS

Our work has three parts: 1) Query refinement that was referred from [7]. Some terms were selected by the genetic algorithms from user profile and used these terms for query expansion. 2) Learning process, wherein the user profile has been updated by using the data feedback from user. The user profile is represented by the equation: $iff_i = iff_i + (c/10)tf_D$ and 3) Document ranking that was shown to user by retrieving from search engine.

5.1 Computation of optimal CIV

Optimal CIV is the process of finding the suitable weight for each class. Genetic algorithm initialized the chromosomes by random value, I:[0,10], for the initial population. We assign population size, *psize*, equal to 30 chromosomes (CIVs) and 50 maximum generations, *maxgen*. In process of genetic algorithm, the reproduction of chromosome will keep the best five chromosomes. Finally, we got the best chromosome to be "optimal CIV". The fitness function for Optimal CIV is

$$sim(u, d_j) = \frac{\vec{u} \bullet \vec{d}_j}{|\vec{u}| \times |\vec{d}_j|}$$

where *u* is user profile and d_j is HTML document from search engine, and $\bullet$ is dot product.

5.2 Optimal CIV and Normal CIV

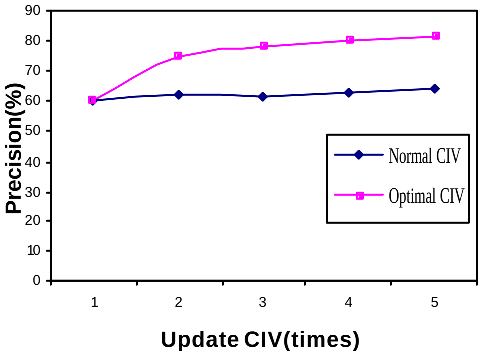
From our experiments, we used precision to compare the performance between Optimal CIV and Normal CIV. The performance would be increased to optimal CIV more than Normal CIV (or TIRS) in percentage. The experiment use three user profiles: "artificial intelligent", "sport car racing" and "health and beauty" as shown in Table 4. We assume that each user profile has no more than 100 terms.

Table 4: The user profiles for testing the system.

User	Title	Optimal CIV
u1	Artificial intelligent	(8,8,6,5,4,3)
u2	Sport car racing	(8,8,5,4,4,3
u3	Health and beauty	(7,8,6,6,3,2)

From Table 4, optimal CIV of a user profile come from genetic algorithm described 5.1. For example, optimal CIV of user profile, "Artificial Intelligent", have class importance value, CIV=(8,8,6,5,4,3), determined class of Title is 8, class of Header is 8, class of Anchor is 6, class of Strong is 5, class of List is 4 and class of Plain Text is 3.

From Figure 1(a) shows the precision values compare with updating optimal CIV each time. The updating of optimal CIV is done when there are user's feedbacks to the system. That is, system update only weight-term in user profile but it does not add any term which is selected from recommend URL into user profile and Figure 1(b) shows the updating of Optimal CIV in each time. The graph shows that the precision value slightly increases while it updated optimal CIV of the order of 3, 4 and 5. The optimal CIV has the best precision value 81.5%.



(a)

Update (times)	Optimal CIV
1	(8,7,8,5,4,3)
2	(8,8,6,5,4,2)
3	(8,8,6,5,3,2)
4	(8,8,6,5,3,2)
5	(8,8,6,5,3,1)

(b)

Figure 1 The precision value of user profile "Artificial Intelligent" in updating CIV.

From Figure 2, when we add new term which is selected, best weight-term value, from recommend URL into user profile. The precision value will increase although we do not update optimal CIV, we used only CIV = (8,8,6,5,4,3). But when the number of terms in user profile increases more than 50, the percentage of precision value will decrease. That is, when there are too many terms in user profile, the system can not specific the significant of user interest.

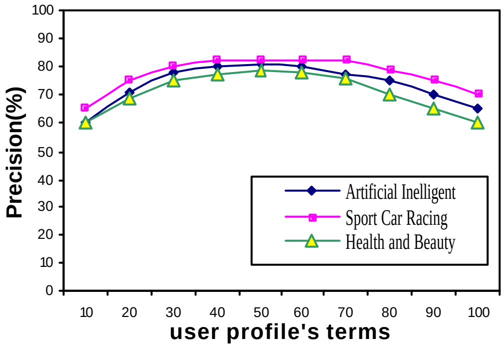
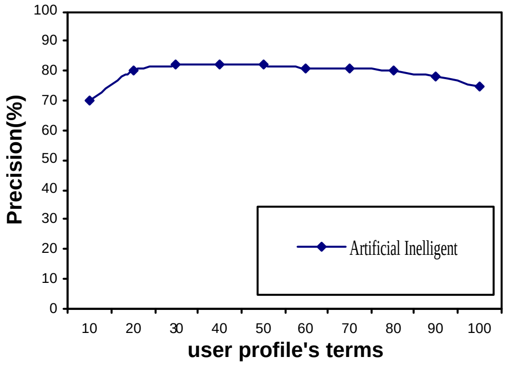


Figure 2 The relationship between precision value and the number of terms in user profile.



(a)

Update (times)	Optimal CIV	Update (times)	Optimal CIV
1	(8,8,6,5,4,3)	6	(8,7,6,5,3,1)
2	(8,8,8,5,4,2)	7	(8,7,6,4,2,2)
3	(8,8,7,4,3,2)	8	(8,7,6,4,3,1)
4	(8,8,7,4,3,1)	9	(8,7,5,3,2,1)
5	(8,8,5,4,3,1)	10	(8,7,5,3,2,1)

(b)

Figure3 The precision value VS. updating Optimal CIV and adding new terms into user profile.

Figure 3(a) shows the graph when the system is added new terms and updated optimal CIV. The system can learn faster than the previous one in Figure 1(a). Figure 3(b) also shows the updating of Optimal CIV each time.

6. CONCLUSION

User profile gives the detail of user's interest. The precision of the system depend on the algorithm that used for maintenance user profile such as finding optimal CIV, adding new terms to user profile and updating weight term in user profile. In our research, we used genetic algorithm to find the optimal value CIV. And we modified the scheme of calculating weight term by considering the HTML tags. From our experiment, terms in class Title and terms in class Header are more important than terms in other classes.

7. REFERENCES

[1] G. Salton and M.J. McGill. *Introduction to Model Information Retrieval*. McGrawHill Book Co., New York, 1983.

[2] G. Salton and M.E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8-36, January 1968.

[3] Z. Zacharis Nick and P. Themis, Web search using a genetic algorithm, *IEEE Internet Computing*, p18-26, March-April 2001.

[4] Cutler M., Shih Y., and Meng W., Using the Structure of HTML Documents to Improve Retrieval, *Proceedings of Usenix Symposium on Internet Technologies and Systems (USITS97)*, Monterey California, December 1997.

[5] Molinari and G. Pasi, A Fuzzy Representation of HTML Documents for Information Retrieval Systems, *in Proc. of the IEEE International Conference on Fuzzy Systems*, New Orleans, 8-12 September 1996.

[6] M. Agosti and A. Smeaton, Information Retrieval and Hypertext, *Kluwer Academic Publishers*, 1996.

[7] S. Phaitoon and P. Ouen, Intelligent Web Search using Genetic Algorithms and User's Profile, *International conference on Artificial Intelligence for the new millemium*, Kota Kinabalu Sabah Malaysia, June 2002.

[8] M.E. Frisse, Searching for Information in a Hypertext Medical Handbook, *Communications of ACM*, 31(7) July 1998.