Evolutionary Learning of Web-Document Structure for Information Retrieval

Sun Kim
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
skim@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@scai.snu.ac.kr

Abstract- Web-documents have a number of tags indicating the structure of documents. The tag information can be utilized to improve the performance of document retrieval systems. In this paper, we propose an approach to retrieve Web-documents using HTML tags and then uses a genetic algorithm to adapt the tag weights. This method uses a modified similarity measure based on the tag weights. A genetic learning method is used to select the tags for retrieval and get the optimal tag weights. We evaluate our method via experiments on conference pages and TREC document sets. The experimental results show that the tag weights are well trained by the proposed algorithm in accordance with the importance factors for retrieval. The proposed method has achieved about 10% improvement in retrieval accuracy.

1 Introduction

The increasing amount of information on the Web raises new and challenging problems for information retrieval [10]. Web search engines have their ancestors in the information retrieval systems developed during the last fifty years. These engines use IR methods such as the Boolean model, the vector space model, the probabilistic model, and the clustering model [1]. However, most of Web-documents are written in HTML (HyperText Markup Language), which consists of tags used for making structure. HTML documents exhibit two kinds of structures not present in plain text documents [3].

1. One is the internal structure consisting of typed text segments marked by HTML tags. HTML defines a set of roles to which text in a document can be assigned. Some of these roles are related to formatting, such as those defining bold and italic text. Others have richer semantic import such as headlines and anchors, the text segments which serve as hyperlinks to other documents.

2. The other is the external structure. As a node in a hypertext, an HTML page is potentially related to a huge number of other pages, through both the hyperlinks it contains and the hyperlinks that point to it from other pages.

Because Web-documents have a kind of structure that plain text documents do not have, the use of HTML structure can be used for the effective retrieval. In this paper, we propose a method for improving retrieval performance using the internal structure of HTML documents. We first present a modified similarity measure using tag weights. After that, A set of tags that are considered to be significant are selected and then the importance factors for the tags are learned for training data using the proposed algorithm. The retrieval for test data is performed using the tags over a threshold value.

For the experiments, we use two data sets, which are Web pages for conference CFP (Call For Paper) and TREC (Text REtrieval Conference) documents. The CFP pages are obtained from the Web to evaluate whether the proposed algorithm learns the tag importance factors properly. The variation of retrieval performance using the tag weights are tested on a TREC document set and the experimental results indicate that our approach can improve the retrieval performance by about 10% on top ranked documents.

The remainder of the paper is organized as follows. In Section 2, we discuss related works. In Section 3, we describe the system framework, and the genetic algorithm for learning tag weights is described in Section 4. Section 5 explains the data set used for the experiments. In Section 6, experimental results are given and conclusions are drawn in Section 7.

2 Related Work

Retrieval systems currently proposed are based on conventional IR models. Most of them ignore hypertext structures as a means of enhancing their retrieval effectiveness. Recent work in information retrieval on the Web is mainly concerned with hyperlink structures (i.e. external structure) [2, 6, 13, 17], not with the tag information. They assume that citations signify deliberate judgment by the page author. For example, if there is a link from page a to page b, then they assume the author of page a recommends page b and links often connect related pages. Spertus [21] observed that cocitation can indicate that two pages are related. That is, if page a points to both pages b and c, then b and c might be related. Chakrabarti et al. [5] use the links and their order to categorize Web pages. They show that the links that are near a given link in page order frequently point to pages on the same topic. An example site using hyperlink information

0-7803-6657-3/01/$10.00 ©2001 IEEE

is Google [4]. Google makes use of the link structure of the Web to calculate a quality ranking (PageRank) for each Web page. A page can have a high PageRank if there are many pages that point to it, or if there are some pages that point to it and have a high PageRank.

Boyan et al. implemented the LASER system, which offers a number of parameters that influence the rankings in produces [3]. The parameters affect how the retrieval function responds to words in HTML fields, how hyperlinks are incorporated, how partial-word matches or query-term adjacency are adjusted and more. Given the parameters, they applied a simulated annealing to optimize the retrieval function.

One of the recent work that uses HTML structures is Cutler et al.'s [7]. The word frequencies are weighted by the tag importance factors, which are obtained using a genetic algorithm. In the paper, training set and test set are not divided and genetic algorithms search the optimal factors on the given queries.

In information extraction from Web pages, it is possible to get increased accuracy by taking advantage of the HTML tag structure. A common approach to extracting information from Web pages is to make site-specific wrappers that extract information based on regularities of the HTML tag structure typically present in a Web site [14].

In information retrieval, genetic algorithms have been used in several ways. An approach for document indexing is presented by Gordon [9]. Competing document descriptions (keywords) are associated with a document and altered by using genetic operations in the approach. A keyword represents a gene and a document's list of keywords represents an individual. A collection of documents initially judged relevant by a user represents the initial population. Based on a fitness measure, the initial population evolves through generations and eventually converges to an optimal population. Yang et al. have developed an adaptive method based on genetic algorithms to modify user queries automatically [24]. They report the effect of adopting genetic algorithms in large databases, the impact of genetic operators, and GA's parallel searching capability. Horng and Yeh [11] propose an approach to automatically retrieve keywords in document retrieval. In the paper, they applied genetic algorithms to tune the weight of retrieved keywords.

Feature selection methods using genetic algorithms for document classification were also developed in [22, 25]. The performance of the classifier and the cost of classification are sensitive to the choice of the features used to construct the classifiers. Genetic algorithms are used to find an optimal feature subset.

3 The Retrieval System

The retrieval engine used in this paper is SCAIR (SCAI Information Retrieval engine) which is built to participate in TREC [20]. SCAIR is based on the vector space model in which both documents and queries are represented as vectors [18]. The components of the vector are keywords extracted from documents or queries. The retrieval system ranks the documents according to the similarities between the documents and the query vector. The higher the value of the similarity measure is, the closer to the query vector the document is. In other words, it assumes that a document, which has a high value of similarity, is relevant to a query. The retrieval system returns a list of documents ordered by similarity in descending manner.

The representation of queries and documents is based on the vector space representation which is commonly used in information retrieval literature. A document or a query is regarded as a set of words which is called terms. A document collection is represented as a term-document matrix which is normally very sparse. For example, a query consists of terms, $T = (t_1, t_2, ..., t_i, ..., t_N)$, where t_i represents the ith term, and N is the number of terms. A query vector is represented by the weight vector, $W = (w_1, w_2, ..., w_i, ..., w_N)$, where w_i represents the weight of the ith term t_i,

3.1 Term Weighting

Term weights are real numbers indicating the significance of terms in identifying a document. If a term does not appear in a document, its corresponding weight in the document vector is zero. The weight of a term in a document is computed by the classical $tf \cdot idf$ scheme [19]. tf (*term frequency*) is the number of times that a term t appears in a document. Intuitively, a higher tf should imply more weight, so the weight of term t should be proportional to tf. idf (*inverse document frequency*) is the inverse of document frequency in the collection that contain t. If more documents contain term t, then t is less significant in the document. Therefore, term weights should be inversely proportional to document frequency.

The weighting scheme w_k is defined as follows:

$$w_k = tf_k \cdot \log\left(\frac{N}{df_k}\right), \quad (1)$$

where w_k is the weight of kth term in the document, tf_k is the frequency of the kth term in the document, N is the total number of documents in the collection, and df_k is the number of documents in which the kth term occurs.

3.2 Similarity Measure Using Tag Weights

The similarity measure is expressed as the inner product of the weight vector and the document vector. There is a necessity of the process to apply HTML tag weights because the documents are written in HTML. Thus two additional processes are added to our approach. One is saving the used tags of each document separately, and the other is applying the weight according to the importance of the tags. The terms, which belong to the specific tags, are indicated during indexing. The similarity measure is modified by adding the tag constant, which is multiplied by the term weight in a document.

```
initialize chromosomes
for g = 1 to gmax
        evaluate all chromosomes by fitness function
        for i = 1 to M
                choose two chromosomes p1, p2
                offspring[i] = crossover(p1,p2)
                offspring[i] = mutation(offspring[i])
        end for
        replace M chromosomes by offsprings
end for
return optimal chromosome
```

Figure 1: Learning algorithm using GA

The modified similarity function $sim(d, q)$ is defined as follows:

$$sim(d, q) = \sum_{k=1}^{n} \alpha_{dk} \cdot w_{dk} \cdot w_{qk}, \qquad (2)$$

where w_{dk} is the weight of the kth term in the document d, w_{qk} is the weight of kth term in the query q, and for all the tags which are determined by term k, α_{dk} is the product of the tag weights. (When a term is not tagged at a document, α is 1.0.) After determining the similarity between documents and a query, a sorted list of documents is produced.

4 Our Genetic Approach

As described above, the characteristic of Web-documents is that an HTML document includes tags for formatting and hyperlinks. In order to make use of the internal structure, we proposed the modified similarity measure using tag weights in Section 3. The main problem in the function does not have any deterministic method which finds the optimal tag weights for document retrieval. In our approach, we use a genetic algorithm to learn the weights of the tags [8] and apply them to Web-document retrieval.

Genetic algorithms are problem solving systems based on the mechanism of natural selection and natural genetics. A solution for a problem is represented as a chromosome. The population is a set of chromosomes. Initial population consists of the chromosomes randomly choosed. In every generation, a new set of artificial creatures (chromosomes) is created using pieces of the fittest of the old. The new set is created by chromosomal operations such as crossover and mutation. While randomized, genetic algorithms are no random walk. They efficiently exploit historical information to move new search points with expected improved performance.

We use a genetic algorithm to tune the weights of HTML tags, with the aim of producing an optimal tag weights. The tag weights are encoded as a chromosome. Real numbers which mean tag weights are used as genes in our approach.

Table 1: Evaluation contingency table

	Retrieved	Not retrieved
Relevant	w	x
Not relevant	y	z

The first step is to generate the initial population, which is made by randomly organized chromosomes and the next step is to modify the chromosomes according to the existing data set. The genetic algorithm for learning the tag weights is shown in Figure 1.

4.1 Representation

A chromosome is defined as a list of tag weights. The definition of a chromosome p is represented as $(p_1, p_2, ..., p_i, ..., p_n)$, where p_i denotes the weight of the tag i, and n is the number of tags to be considered. Each gene represents a tag weight. The genes of initial chromosomes are generated randomly and the range of weight values is from 0.0 to 4.0.

4.2 The Fitness Function

The fitness function measures the performance of the retrieval results using tag weights.

When binary scales are used for both relevance and retrieval, a table can be established showing how the document set is divided by these two measures (Table 1). Several measures such as precision, recall are used to evaluate the performance of document retrieval. Precision P is defined as the proportion of retrieved documents that are relevant, $P = \frac{w}{w+y}$. Recall is defined as the proportion of relevant documents that are retrieved, $R = \frac{w}{w+x}$.

The fitness function used to evaluate a chromosome is the non-interpolated average precision, which is one of the evaluation methods at TREC [23]. The non-interpolated average precision is the single-valued measure that reflects the performance over all relevant documents. The average precision for a single topic is the mean of the precision obtained after each relevant document is retrieved. If a relevant document is not retrieved at all, its precision is assumed to be 0. The non-interpolated average precision for a run consisting of multiple topics is the mean of the average precision scores of each of the individual topics in the run. For example, consider a query that has four relevant documents which are retrieved at ranks 1, 2, 4, and 8. The actual precision obtained when each relevant document is retrieved is 1, 1, 0.75, and 0.5, respectively, the mean of which is 0.81. Thus, the non-interpolated average precision for this query is 0.81. Geometrically, non-interpolated average precision is the area underneath a non-interpolated recall-precision curve.

4.3 Genetic Operators

The genetic algorithm uses crossover and mutation operators to generate the offspring of the existing population. Before genetic operators is done, parents should be selected for evolution to the next generation. In our approach, the parents are selected randomly from a half of the population in the decreasing order of quality. The quality of a chromosome is determined by the fitness function.

The selected parents produce offspring by crossover. The crossover used is the arithmetical crossover, which assigns the average of two parents for each location to the corresponding location of the offspring [15]. Let p_x and p_y be two selected parent chromosomes, which are represented respectively as follows:

$$p_x = (p_{x1}, p_{x2}, p_{x3}, ..., p_{xn})$$

$$p_y = (p_{y1}, p_{y2}, p_{y3}, ..., p_{yn}).$$

Let p_z be the offspring generated:

$$p_z = (p_{z1}, p_{z2}, p_{z3}, ..., p_{zn}).$$

The weight vectors of the offspring are defined as follows:

$$p_{zi} = \frac{\beta(p_{xi} + p_{yi})}{2.0}, \quad (3)$$

where

$$\beta = random(0, 1),$$

where *random* is a function to generate randomly an integer number within the range of the values with probability P_c.

The mutation is done for variety after crossover. It changes the value of randomly selected position in a random chromosome with probability P_m. A half of the population other than the selected parents are substituted by the produced offspring.

5 Data Sets

We now describe the data set used for the experiments. For the proposed method, we use CFP and TREC data set. CFP documents are collected for evaluating whether the genetic algorithm finds the tag weights according to the importance of document retrieval and TREC document set is used to evaluate the effect of retrieval using HTML internal structure.

5.1 CFP Data

CFP data is documents from the Web, which collects the conference pages including SIGIR, IJCAI, EuroGP, and so on. Two data sets, Set A and Set B, are constructed for the experiments. Each set includes 100 documents and has same contents. The difference between Set A and Set B is the internal structure, which is designed using HTML tags. Original 100 documents obtained from the Web are included in Set A. Set B includes modified documents from Set A. The keywords of each document in Set B are emphasized by header tag (<H>). Figure 2 shows document examples of Set A and Set B. 'Information' and 'Retrieval' which are keywords in the document are embedded in header tag (<H3>) in Set B.

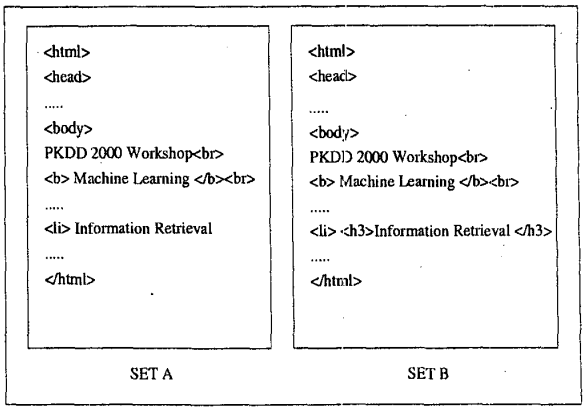

Figure 2: Sample CFP documents

<title> foreign minorities, Germany
<desc> Description:
What language and cultural differences impede the integration of foreign minorities in Germany?
<narr> Narrative:
A relevant document will focus on the causes of the lack of integration in a significant way; that is, the mere mention of immigration difficulties is not relevant. Documents that discuss immigration problems unrelated to Germany are also not relevant.

Figure 3: A sample TREC topic

For queries, one query consisting of six words, 'genetic', 'algorithms', 'conference', 'specially', 'information' and 'retrieval' was generated. It assumes that user want to find the conferences related to genetic algorithms, specially including information retrieval section. Intuitively, the header tag in Set B should get more weight than other tags after trained by the proposed algorithm.

Relevant documents of the query are manually determined. While 10 documents are judged as relevant, others are irrelevant.

5.2 TREC Data

The documents for TREC data is WT2g, which is used to Web Track of TREC sponsored by NIST (National Institute of Standards and Technology) [16]. It was collected by Internet Archive and includes all WWW pages [12]. There are 247,491 distinct pages of 2 Gigabytes size.

A query is called a topic in TREC. A topic consists of ti-

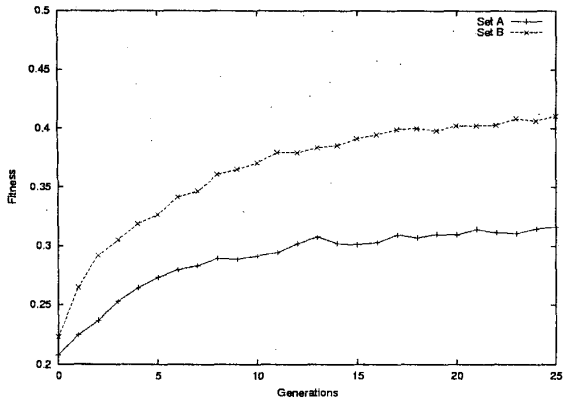

Figure 4: Average fitness for generations on CFP data

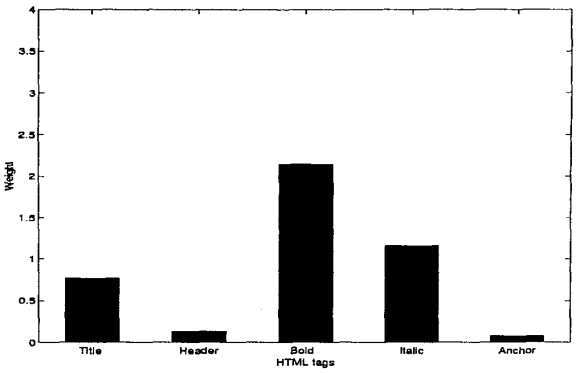

Figure 5: HTML tag weight averages for Set A

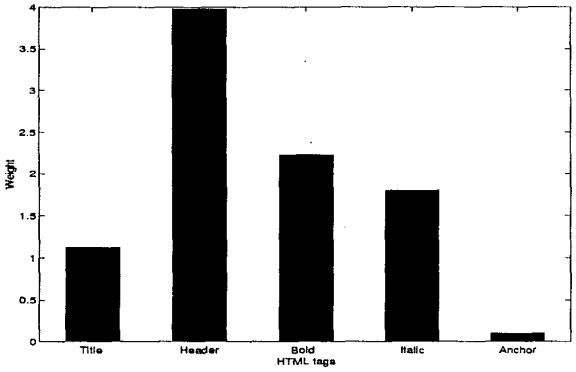

Figure 6: HTML tag weight averages for Set B

tle, description, and narrative. A sample topic is shown in Figure 3. The title has been specially designed to allow experiments with very short queries. The title consists of up to three words that best describe the topic. The description field is one sentence description of the topic area. The description field contains all of the words in the title field. The narrative gives a concise description of what makes a document relevant.

The title and description field of a topic are considered as queries in the experiments. 10 queries (Topic No. 401 to 410) are used for learning phase and another 10 queries (Topic No. 411 to 420) are used for retrieval.

We use the results of relevant documents for queries, which are published by NIST. Relevant documents are judged using the pooling method [23, 26]. In this method, a pool of possible relevant documents is created by taking a sample of documents selected by the various participating systems of TREC. This pool is then shown to the human assessors. The sampling method used in TREC is to take the top 100 documents retrieved per a judged run for a given topic and merge them into the pool for assessment. This is a valid sampling technique since all the systems use the ranked retrieval methods, with those documents most likely to be relevant returned first. Each pool is sorted by document ID, so that assessors cannot tell if a document was highly ranked by some system or how many systems retrieved the document.

6 Experimental Results

In this section, we present our experiments on learning tag weights and retrieval using tag weights. The CFP data is used to judge whether the genetic learning gets the tag weights correctly, and TREC data is used to show the change when the tag weights learned by the proposed algorithm are applied for document retrieval.

Five HTML tags (<TITLE>, <H>, <B>, <I> and <A>) are used for the experiments. They mean Title, Header, Bold, Italic and Anchor, respectively. The Title and Header use only the words in a Web page marked as a part of the title or headings to classify that page. Thus, words in the tags are as a representatives of the page. Bold and Italic are taken because they are used to emphasize words. We assume that the hyperlinks of a document generally are connected to the related documents. The anchor, which links to another document, is added to the tags by the assumption.

For each experiment, learning is repeated 20 times until the 30 generations with 100 population size.

6.1 Experiment 1: CFP Data

The approach developed in this study was performed on CFP data with Set A and Set B. The retrieval system returns one hundred documents for a topic ordered by the similarities between documents and a query.

Experimental results in Figure 4 indicate that the average fitness for generations with Set A and Set B. Set B is better structured than Set A by tagging the keywords in documents using header. As shown in the result, Set B shows higher precision than Set A, and the difference between Set A and Set B become greater as the generation continues.

The chromosome that has the highest fitness over genera-

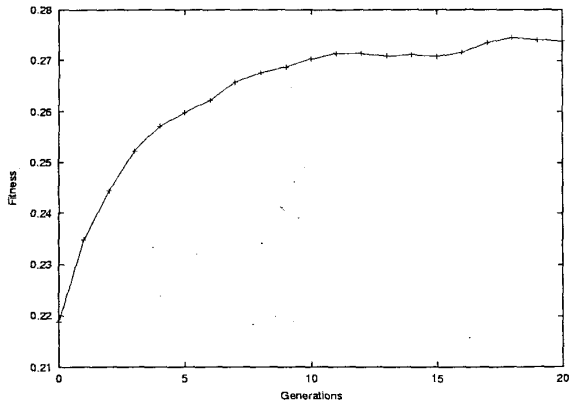

Figure 7: Average fitness for generations on TREC data

Table 2: HTML tag weights averages

HTML tag	Weights
Title	0.6
Header	1.6
Bold	0.7
Italic	0.6
Anchor	1.6

tions is regarded as the optimal tag weights. The tag weights learned by genetic algorithm are shown in Figure 5 and 6. For the Title, Header, Bold, Italic, and Anchor, the average weights are 0.77, 0.13, 2.15, 1.16, and 0.08, respectively in Set A. For Set B, they are 1.12, 3.98, 2.23, 1.80 and 0.10. The tag weights except header, which has higher value in Set B than Set A, are similar both Set A and Set B. As a result, it shows that the proposed learning algorithm works efficiently as mentioned above.

6.2 Experiment 2: TREC Data

For the TREC documents, 10 queries of TREC-8 are selected for training and other 10 queries are for retrieval. The results returned by the system are limited to two hundred documents for a topic because users usually want to receive a small number of documents for a query, i.e. they want high precision, and low recall.

The steps used in retrieval using tag weights are given below:

1. Adapt tag weights for training queries at given times.
2. Select the optimal tag weights by considering the chromosome which has the highest fitness.
3. Select only the tags which have weights over 1.0 at the chromosome.
4. Retrieve 200 documents per topic not using tag weights.
5. Adjust the similarity of each retrieved document with the selected tag weights.
6. Rerank the retrieved documents by ordering for new similarities.

Table 3: Comparison of non-interpolated average precision

	Average precision
Without tag weights	0.2383
With tag weights	0.2503

Table 4: Comparison of precision at N retrieved documents

	P@10	P@20
Without tag weights	0.3100	0.3750
With tag weights	0.3400	0.4050

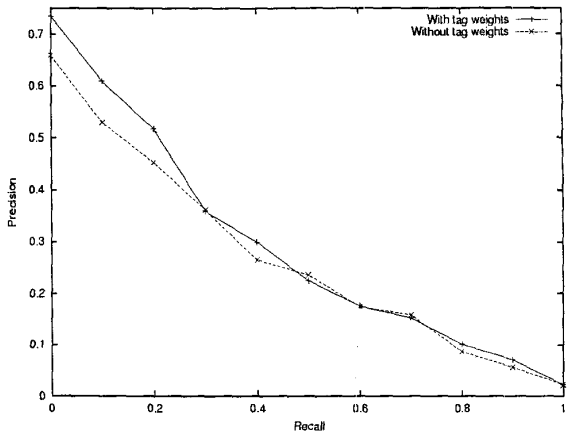

Figure 8: Interpolated recall-precision averages

As the generation continues, further improvement is found in average population fitness as demonstrated in Figure 7. The fitness rapidly increases until about 8th generation. After then, the fitness increases slowly, which is caused by the arithmetic crossover. The offspring is generated by the average of the parents which have high fitness. In addition to it, the substituted chromosomes are the half of the population in one generation. Therefore, the early generations are converged to the chromosomes of the population which have high fitness even after one generation. As the generation progresses a little more, the increment of fitness falls down a little because most of chromosomes were already converged to the high fitness chromosomes.

The average of selected weights is presented in Table 2. For the Title, Header, Bold, Italic and Anchor, the average weights are 0.6, 1.6, 0.7, 0.6 and 1.6 respectively.

The comparison of our method with the method not using tag weights is shown in Table 3 and 4. The experimental results in the tables indicate that using tag weights give better performance than normal retrieval in the precision average.

Table 5: Comparison of average precision for recall. Precision at recall 0.1 is taken to be maximum of precision at all recall points ≥ 0.1.

Recall	Without tag weights	With tag weights
0.0	0.6574	0.7350
0.1	0.5284	0.6081
0.2	0.4514	0.5158
0.3	0.3601	0.3580
0.4	0.2645	0.2990
0.5	0.2339	0.2227
0.6	0.1735	0.1745
0.7	0.1583	0.1525
0.8	0.0862	0.1007
0.9	0.0561	0.0705
1.0	0.0219	0.0219

Figure 8 shows the precision-recall curves for using and not using the tag weights. When the recall is under 0.3, the retrieval performance using tag weights is higher than not using tag weights. High precision at low level of recall means that there are more relevant documents on top ranked documents. Table 5 describes the average precision as recall increases. The retrieval results with tag weights have high precision when recall is low.

7 Conclusions

This paper proposes an approach that uses the internal structure of HTML documents to improve retrieval performance. Genetic algorithms, which are generally quite effective for rapid global search in large search spaces, are applied to find the optimal HTML tags. The approach is used to retrieve CFP and TREC documents according to the tag weights learned.

According to our experiments, the document retrieval which takes advantage of HTML tags performs better than the traditional IR approach which uses plain texts. We find that the use of HTML structures gives an effect on the retrieval performance and the performance depends on how the well HTML documents are structured.

It is interesting to note that the results show high precision at low recall. Generally, users do not need many relevant documents in all retrieved documents. They are satisfied with finding relevant documents at top ranked documents. Our experimental results show the use of HTML tags for effective retrieval of Web-documents, especially for documents that are well-structured.

However, we also find that improvement is limited to some extent, because HTML documents are usually semi-structured. In the future, we plan to design such techniques that will make HTML documents well-structured or that will convert Web-documents into well-structured documents. In addition, XML document retrieval using our approach will be evaluated.

Acknowledgments

This research was supported in part by AITrc, the Korea Ministry of Information and Telecommunications under Grant 00-034 through IITA, and the Ministry of Education under the BK21-IT Program.

Bibliography

[1] Belkin, N. J. and Croft, W. B., Retrieval Techniques. *Annual Review of Information Science and Technology* 22, pp. 109–145, 1987.

[2] Bharat, K. and Henzinger, M. R., Improved Algorithms for Topic Distillation in a Hyperlinked Environment, *Proceedings of the ACM SIGIR'98 Conference*, pp. 104–111, 1998.

[3] Boyan, J., Freitag, D. and Joachims, T., A Machine Learning Architecture for Optimizing Web Search Engines, *Proceedings of the AAAI workshop on Internet-Based Information Systems*, pp. 1–8, 1996.

[4] Brin, S. and Page, L., The Anatomy of a Large-scale Hypertextual Web Search Engine, *The Seventh International World Wide Web Conference (WWW7)*, 1998.

[5] Chakrabarti, S., Dom, B., Gibson, D., Kumar, S. R., Raghavan, P., Rajagopalan, S and Tomkins, A., Experiments in topic distillation, *ACM-SIGIR '98 Post-Conference Workshop on Hypertext Information Retrieval for the Web*, 1998.

[6] Chakrabarti, S., Data Mining for Hypertext: A Tutorial Survey, *ACM SIGKDD Explorations* 1 (2), pp. 1–11, 2000.

[7] Cutler, M., Deng, H., Maniccam, S and Meng, W., A New Study on Using HTML Structures to Improve Retrieval, *The Eleventh IEEE Conference on Tools with AI*, 1999.

[8] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[9] Gordon, M., Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM* 31, pp. 1208–1218, 1988.

[10] Gordon, M. and Pathak, P., Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines, *Information Processing & Management* 35 (2), pp. 141–180, 1999.

[11] Horng, J.-T. and Yeh, C.-C., Applying Genetic Algorithms to Query Optimization in Document Retrieval, *Information Processing and Management* 36, pp. 737–759, 2000.

[12] Internet Archive, *Building an Internet Library*, http://www.archive.org.

[13] Kleinberg, J., Authoriatative Sources in a Hyperlinked Environment, *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*, pp. 668–677, 1998.

[14] Kushmerick, N., Weld, D. S., and Doorenbos, R., Wrapper Induction for Information Extraction, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 729–735, 1997.

[15] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer, pp. 104–105, 1992.

[16] NIST, *Text REtrieval Conference homepage*, http://trec.nist.gov.

[17] Picard, J., Modeling and Combining Evidence Provided by Document Relationships Using Probabilistic Argumentation Systems, *Proceedings of the ACM SIGIR'98 Conference*, pp. 182–189, 1998.

[18] Salton, G., Wong, A. and Yang, C. S., A Vector Space Model for Automatic Indexing, *Communications of the ACM* 18, pp. 613–620, 1975.

[19] Salton, G., *Automatic Text Processing*, Addison-Wesley, pp. 279–281, 1989.

[20] Shin, D. H. and Zhang, B. T., A Two-Stage Retrieval Model for the TREC-7 Ad Hoc Task, *The Seventh Text Retrieval Conference (TREC-7)*, 1998.

[21] Spertus, E., ParaSite: Mining Structural Information on the Web, *The Sixth International World Wide Web Conference (WWW6)*, 1997.

[22] Tseng, L. Y. and Yang, S. B., Genetic Algorithms for Clustering, Feature Selection and Classification, *International Conference on Neural Networks* Vol. 3, pp. 1612–1616, 1997.

[23] Voorhees, E. M. and Harman, D., Overview of the Eighth Text Retrieval Conference, *The Eighth Text Retrieval Conference (TREC-8)*, 1999.

[24] Yang, J., Korfhage, R. R. and Rasmussen, E., Query Improvement in Information Retrieval using Genetic Algorithms: A Report on the Experiments of the TREC Project, *The First Text Retrieval Conference (TREC-1)*, 1993.

[25] Yang, J. and Honavar, V., *Feature Extraction, Construction and Selection - A Data Mining Perspective*, Kluwer Academic Publishes, pp. 117–136, 1998.

[26] Zobel, J., How Reliable are the Results of Large-Scale Information Retrieval Experiments?, *Proceedings of the ACM SIGIR'98 Conference*, pp. 307–314, 1998.