Web-Document Retrieval by Genetic Learning of Importance Factors for HTML Tags

Sun Kim and Byoung-Tak Zhang

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

Abstract. In contrast to conventional documents, a Web document consists of a number of tags which provide hints on the structure of the documents. In this paper, we propose a Web-document retrieval method using the characteristics of HTML tags. This method learns the importance of tags from a training text set. We use a genetic algorithm for learning the importance weights. We also present a modified similarity measure which uses the tag information. Experiments have been performed on the TREC document collection consisting of 247,491 documents. Compared to the traditional IR method, the proposed method has achieved 15% improvement in average precision.

1 Introduction

The World Wide Web is revolutionizing the way that people access information. The amount of available information on the Web is increasing rapidly [10]. As the amount of information on the World Wide Web grows, it becomes increasingly difficult to find what we want. There are a number of Web search engines which retrieve the URLs for the documents of users' interests. These engines use statistical methods for indexing plain text documents. However, most of Web documents are written in HTML (HyperText Markup Language). HTML documents consist of two kinds of structure not present in plain text documents [2].

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, a HTML page is related to potentially huge numbers of other pages, through both the hyperlinks it contains and the hyperlinks that point to it from other pages.

A.-H. Tan and P. Yu (Eds): PRICAI 2000 Workshop on Text and Web Mining, Melbourne, pp. 13-23, August 2000.

In a traditional IR approach, only the words appearing in a document are considered as the elements for retrieval. In this paper, we propose a method for improving the retrieval performance using the internal structure of HTML documents. We first select a set of tags that is considered to be significant. Next, the importance factors for the tags are learned using a genetic algorithm. To carry out the experiments, we used the queries and document set of TREC (Text REtrieval Conference) [12]. The experiments indicate that our approach can improve the retrieval performance by about 15% on top ranked documents.

This paper is organized as follows. In Section 2, we discuss the related work. Section 3 describes the retrieval system and the method for learning tag information is described in Section 4. Section 5 explains the data set used for the experiments. Experimental results are given in Section 6.

2 Related Work

Recent work in information retrieval on the Web is mainly for hyperlinks (i.e. external structure) [1, 13, 21], not for the tag information. They assume that if there is a link from page a to page b, then the author of page a recommends page b and links often connect related pages. Spertus [18] observed that co-citation 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. [4] use the links and their order to categorize Web pages and 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 is Google [3]. 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 [2]. The parameters affect how the retrieval function responds to words in HTML fields, how hyperlinks are incorporated, how to adjust for partial-word matches or query-term adjacency and more. Given the parameters, they applied a simulated annealing to optimize the retrieval function.

One of the recent work that is similar to our approach is Cutler et al.'s [5]. They used HTML structures to improve retrieval performance. A genetic algorithm is used to find the optimal tag importance factor. They used a small document set which consists of 3,040 distinct pages. The main drawback of this paper is that same queries are used for both learning and retrieval, which causes the tag importance factor to be overfitted to that queries.

In information retrieval, genetic algorithms have been used in several ways. An approach for document indexing was presented by Gordon [7]. Competing document descriptions (keywords) are associated with a document and altered by using genetic operations in the approach. A keyword presents 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 evolved through generations and eventually converged to an optimal population. Yang et al. have developed an adaptive method based on genetic algorithms to modify user queries automatically [22]. They reported the effect of adopting genetic algorithms in large databases, the impact of genetic operators, and GA's parallel searching capability. Feature selection methods for document classification were also developed in [19, 23]. 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 Retrieval System

The retrieval engine used in this paper is SCAIR (SCAI Information Retrieval engine) which is built to participate in TREC competition [17]. SCAIR is based on the vector space model [15]. A document is regarded as a set of words. If one word is a term, a document is represented as a list of terms or vector. A document collection is represented as a term-document matrix which are normally very sparse. A query is also represented as a list of terms or a term vector.

Documents are indexed by the classical $tf \cdot idf$ weighting scheme [16]:

$$w_{ik} = tf_{ik} \cdot \log\left(\frac{N}{df_k}\right), \tag{1}$$

where w_{ik} is the weight of kth term in the ith document, tf_{ik} is the frequency of the kth term in the ith 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.

Query terms are weighted only by the idf value, where idf is represented by $\log(\frac{N}{df_k})$. The similarity between a query and a document is measured by modified cosine coefficient:

$$sim(d_i, q_j) = \frac{\sum_{k=1}^{n} \alpha_{ik} \cdot w_{ik} \cdot q_{jk}}{\sqrt{\sum_{k=1}^{n} (\alpha_{ik} \cdot w_{ik})^2 \cdot \sum_{k=1}^{n} q_{jk}^2}}, \tag{2}$$

where w_{ik} is the weight of the kth term in the ith document, q_{jk} is the weight of kth term in the jth query, and for all the tags which are determined by term k, α_{ik} is the product of the tag weights. After determining the similarity between the document and the query, a sorted list of documents is produced.

On the other hand, 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 for 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 tag weights are applied to evaluate the similarity. It is represented as a constant α in the equation (2).

```
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
```

Fig. 1. Learning algorithm using GA

4 Learning Tag Importance Using a Genetic Algorithm

As described above, the characteristic of Web documents is including tags for formatting and hyperlinks. There could be many methods in learning the weights of the tags. In our approach, we used the genetic algorithm to learn the weights of the tags and applied them to Web-document retrieval [6].

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 [8]. While randomized, genetic algorithms are no random walk. They efficiently exploit historical information to move new search points with expected improved performance.

In our approach, a chromosome represents tag weights and consists of a list of real number. The initial population is made by randomly organized chromosomes. The selection of parents for evolution to the next generation is as follows. 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 fitness function measures the performance of the retrieval results about the tag weights. The 11-point average precision is used for the fitness value, which is one of the evaluation methods at TREC [20].

The selected parents produce offspring by crossover. The crossover used is the arithmetical crossover, which assigns the average of two parents for each location

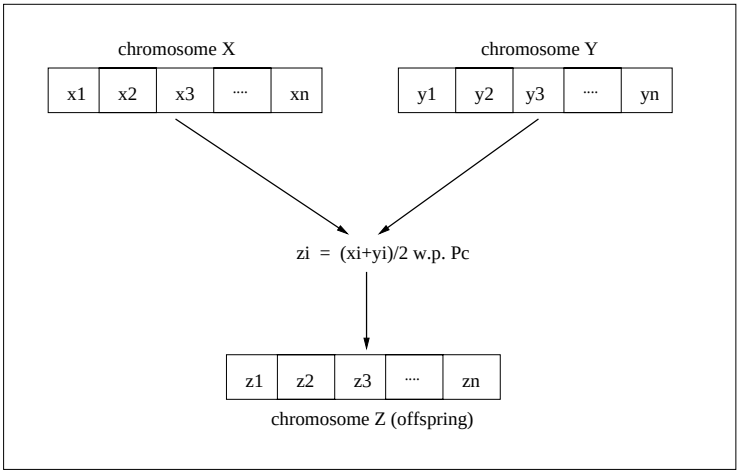

Fig. 2. Crossover

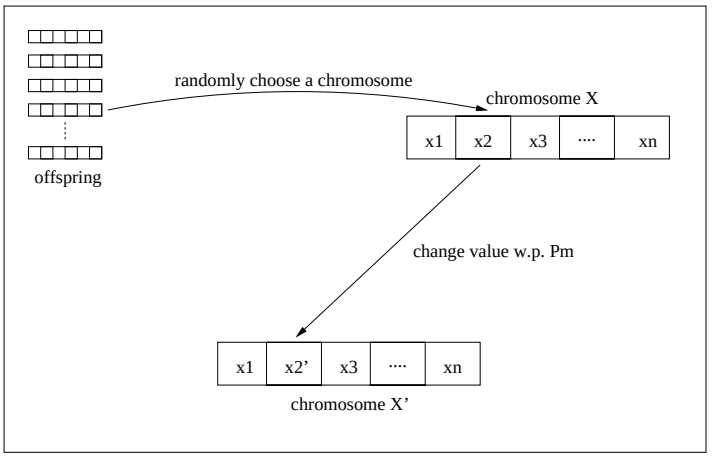

Fig. 3. Mutation

to the corresponding location of the offspring [11]. A half of the population other than the selected parents are substituted by the produced offspring. The mutation is done for variety after crossover. It changes the value of randomly selected position in a random chromosome. The genetic algorithm for learning the tag weights is shown in Figure 1. Figure 2 and Figure 3 describe crossover and mutation.

<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.

Fig. 4. A sample TREC topic

5 Data Set

The document set for the experiments is WT2g, which is used to Web Track of TREC [12]. It was collected by Internet Archive and includes all WWW pages [9]. There are 2 Gigabytes, 247,491 distinct pages.

A query is called a topic in TREC. A topic consists of title, description, and narrative. A sample topic is shown in Figure 4. The titles have been specially designed to allow experiments with very short queries. The titles consist 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.

Relevant documents are judged using the pooling method [20]. 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.

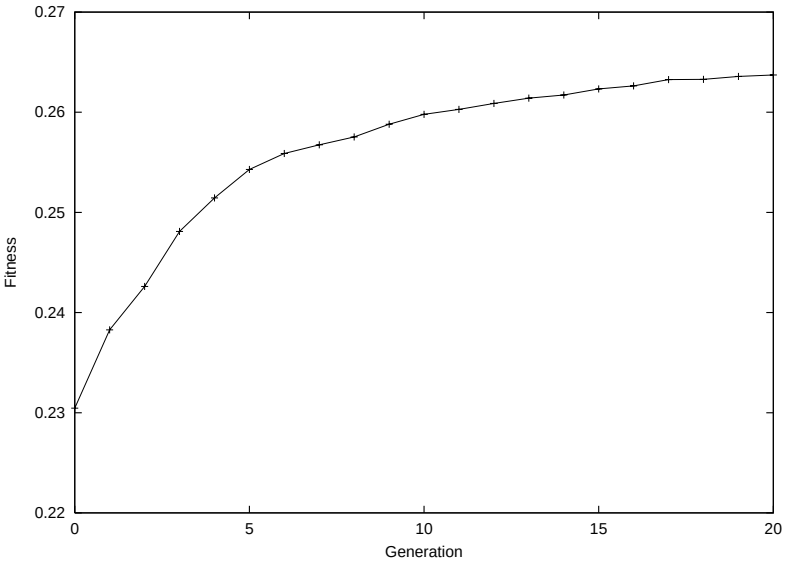

Fig. 5. Average fitness for generations

6 Experimental Results

Experiments were conducted to find optimal tag weights. The title and description field of a topic were considered as queries. 10 queries (Topic No. 401 to 410) were used for learning and another 10 queries (Topic No. 411 to 420) were used for retrieval. The retrieved documents for a topic are ten hundreds ordered by the similarity between document and query. The evaluation is average precision in information retrieval.

Five tags (<TITLE>, <H>, <B>, <I> and <A>) were 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 as headings to classify that page. Thus, words in the tags were taken as representative of a page. Bold and Italic were taken because they are used to emphasize words. We assumed that the hyperlinks of a document generally are connected to the related documents. The anchor, which links to another document, was added to the tags by the assumption.

The learning was repeated 20 times. The used parameters are the following:

- Population size: 100
- Number of generation: 25
- Probability of mutation: 0.04
- Range of weight: $0.0 \sim 4.0$

As the generation continues, further improvement is found in average population fitness as demonstrated in Figure 5. The fitness rapidly increases until about 8th generation. After then, the fitness increases slowly. It is caused by the arithmetic crossover. The offspring are 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 becomes slack because most of chromosomes were already converged to the high fitness chromosomes closely.

Table 1. HTML tag weights averages

HTML tag	Weights
Title	0.5584±0.2822
Header	2.3425±0.2614
Bold	0.7060±0.2061
Italic	1.0192±0.3128
Anchor	1.7634±0.1306

Table 2. Comparison of the results between tag and no tag information

	Relevant documents	11-point average precision
Without tag information	262	0.2325
With tag information	252 ± 2	0.2367±0.0045

The chromosome that has the highest fitness over generations was regarded as the optimal tag weights. The top five chromosomes are selected for retrieval. The averages of selected weights are presented in Table 1. For the Title, Header, Bold, Italic and Anchor, the average weight is 0.5584, 2.3425, 0.7060, 1.0192 and 1.7634, respectively.

Table 2 represents the number of relevant documents and the 11-point average precision for retrieval. There hardly exists any difference on the number of relevant documents when the weight is applied. But, average precision is improved.

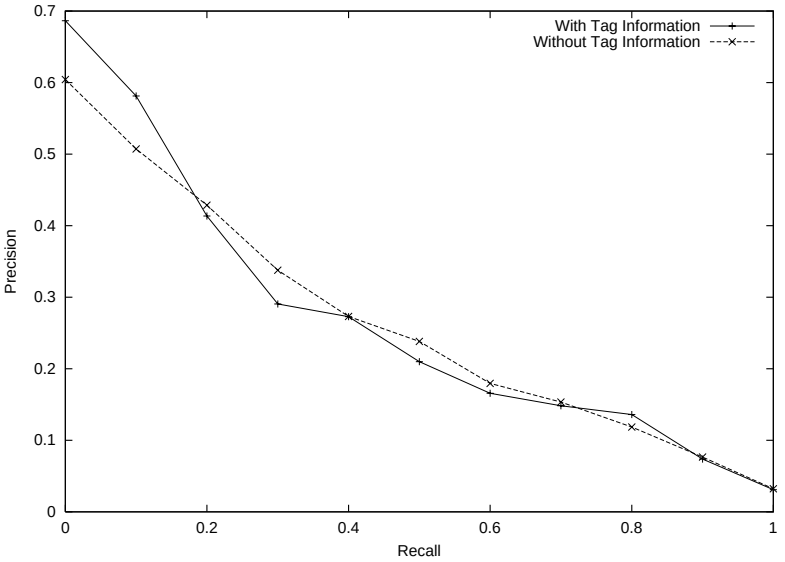

Fig. 6. Interpolated recall-precision averages

Table 3. 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 information	With tag information
0.0	0.6042	0.6865±0.0287
0.1	0.5073	0.5813±0.0235
0.2	0.4286	0.4135±0.0119
0.3	0.3377	0.2906±0.0227
0.4	0.2731	0.2728±0.0249
0.5	0.2382	0.2099±0.0060
0.6	0.1795	0.1657±0.0064
0.7	0.1534	0.1483±0.0047
0.8	0.1186	0.1360±0.0025
0.9	0.0766	0.0738±0.0013
1.0	0.0323	0.0309±0.0002

Some results show that the average precision is lower than retrieval result without tag information. It is caused by that the weights are overfitted to the training data.

Figure 6 shows the precision-recall curves for using and not using the tag weights. When the recall is under 0.2, the retrieval performance using tag information is higher than not using tag information. High precision at low levels of recall means that there are more relevant documents in top documents. Table 3 describes the average precision as recall increases. The retrieval results with tag information have high precision when recall is low.

7 Conclusions

In this paper, we proposed an approach that uses the HTML tag weights to improve retrieval performance. A genetic algorithm is used to find the optimal tag weights. Genetic algorithms are generally quite effective for rapid global search in large search spaces.

According to our experiments, the retrieval which takes advantage of HTML structure performs better than the traditional IR approach which uses plain texts. We found that Header and Anchor provide useful information to improve the retrieval performance.

It is interesting to note that there exist hardly differences when only relevant documents are considered. However, the results show high precision at low recall. Generally, the 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 importance of the tag information for effective retrieval of Web-documents.

Further experiments are needed to check if our approach will improve the retrieval performance when it uses expanded topics. Learning the weights for more HTML tags will be conducted to find the importance. Furthermore, the problem that the population is converged to local minimum, i.e. overfitting to training data has to be solved.

Acknowledgements

This research was supported by the Korea Ministry of Information and Telecommunications under Grant C1-98-0068-00 through IITA.

References

1. 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.
2. 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.

3. Brin, S. and Page, L., The Anatomy of a Large-scale Hypertextual Web Search Engine, *The Seventh International World Wide Web Conference(WWW7)*, 1998.
4. 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.
5. 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.
6. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
7. Gordon, M., Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM* 31, pp. 1208-1218, 1988.
8. Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
9. Internet Archive, *Building an Internet Library*, http://www.archive.org.
10. Lawrence, S. and Giles, C. L., Searching the World Wide Web, *Science*, Vol. 280, pp. 98-100, 1998.
11. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolutionary Programs*, Springer, pp. 104-105, 1992.
12. NIST, *Text REtrieval Conference homepage*, http://trec.nist.gov.
13. 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.
14. Raghavan, V. V. and Argarwal, B., Optimal Determination of User-Oriented Clusters: An Application for the Reproductive Plan, *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, pp. 241-246, 1987.
15. Salton, G., Wong, A. and Yang, C. S., A Vector Space Model for Automatic Indexing, *Communications of the ACM* 18, pp. 613-620, 1975.
16. Salton, G., *Automatic Text Processing*, Addison-Wesley, pp. 279-281, 1989.
17. 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.
18. Spertus, E., ParaSite: Mining Structual Information on the Web, *The Sixth International World Wide Web Conference(WWW6)*, 1997.
19. 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.
20. Voorhees, E. M. and Harman, D., Overview of the Eighth Text Retrieval Conference, *The Eighth Text Retrieval Conference(TREC-8)*, 1999.
21. Weiss, Ron., Vélez, B. and Sheldon, M. A., HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering, *Proceedings of the Seventh ACM Conference on Hypertext*, pp. 180-193, 1996.
22. 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.
23. Yang, J. and Honavar, V., *Feature Extraction, Construction and Selection - A Data Mining Perspective*, Kluwer Academic Publishes, pp. 117-136, 1998.