Search Improvement by Genetic Algorithms with a Semiotic Network

Sang-yon Lee, Sung-Soon Choi and Byung-Ro Moon
School of Computer Science and Engineering,
Seoul National University, Seoul, Korea
{slee,irranum,moon}@soar.snu.ac.kr

Abstract

The explosive expansion of the World Wide Web makes the search problems more challengeable. In this paper we present a search improvement method based on a semiotic connection network and a genetic algorithm. The semiotic connection network expands given keywords to an extended set of keywords. The genetic algorithm tunes up the parameters for search. The experimental results showed 6% improvement over Google's search results. The proposed method was incorporated into a commercial product.

1 Introduction

The quantity of available information in the World Wide Web (WWW) is explosively growing. Effective search became crucial due to the huge volume of information. To date, diverse search methods and exploring agents have been presented.

Since 1994 [12][13][18], lots of Internet search engines have been developed; some of them have been commercially utilized. Early Internet search methods were to find the documents containing requested word strings in a bunch of documents collected from the WWW by crawlers.

However, simple Internet search methods like word-string matching turned out to have problems. With the sharp increase of Web pages, they tended to show a considerable number of useless URLs (Uniform Resource Locators) rather than the Web pages that users want.

A notable approach is to evaluate Web pages from the view point of text documents. There were studies that represent each document as a vector of words included in the document and evaluate documents with the similarities between the vectors [8]. Another notable approach is extended-word method; when a user provides a query, it extends the query by generating additional words [5].

A new approach exploits the fact that Web pages are hypertext documents containing tags such as links and anchors. It evaluates the importance of each Web page with the number of backward links. The Web pages having more incoming links are thought to be more important pages [17]. This approach produced a famous search engine Google [4]. This method includes counting the incoming and outgoing links of a Web page; it led to the study that models the entire WWW structure as a graph, by conceptualizing URLs and links as nodes and edges, respectively [11].

There were studies with stochastic optimization methods for Internet search engines. In [21], genetic algorithm (GA) was used for tuning up parameters of a Web agent for information retrieval. In [22], simulated annealing was used for an Internet search engine.

The link-based analysis exploits the fact that the importance of a Web page depends on the number of incoming links from other Web pages. This method puts emphasis upon the connections of Web pages. It is efficient for finding popular Web sites. However, if a user wants to find a specific content directly, a Web page with only a lot of backward links would not be satisfactory. Besides, it is sometimes not easy to evaluate the relative importance of the documents with few incoming links.

On the other hand, although the content-based analysis exploited various analytic methods, the vagueness of evaluating the importance of documents makes the evaluation difficult. Among many elements present in a document, it is not easy to clarify which elements are important and close to the themes that a user wants to find.

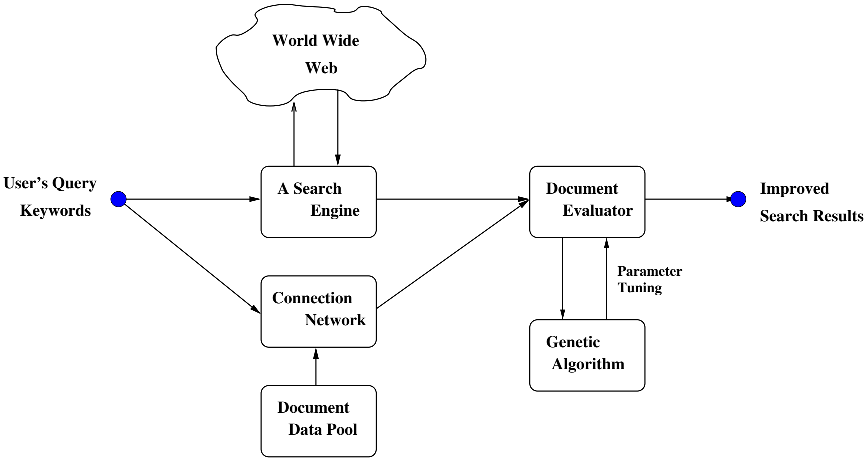

Figure 1: Operational frameworks

In this paper, we suggest a number of elements for document evaluation and optimize them using a genetic algorithm. Our primary purpose is to improve users' satisfaction with search engines.

Semiotics is a general theory of signs and symbols including the analysis of the nature and relationship of signs. The document analysis based on semiotics utilizes the relationship of words appearing in documents. Rather than thinking of each word independently, it takes a relational viewpoint [7]. We reflect the semiotic relationship of words into a connection network and use it for genetic evaluation of documents. The system uses GA as a method of evolutionary optimization to tune up the ranking factors deciding the relative importance of Web pages

The rest of this paper is organized as follows. In section 2 the architecture of the system and the data for experiments are presented. Section 3 deals with the methodology on how to use the connection network to expand keywords and how to use TF-IDF (Term Frequency, Inverse Document Frequency) for document evaluation. In section 4, we describe the parameter tuning by genetic algorithms. In section 5, we give our experimental results and compare our results with those of Google. Finally, the conclusion is given in section 6.

2 System Architecture

The purpose of this study is to find a method to improve the results of search engines. We designed the system to begin with the results of existing search engines. For experiments, the system was designed as a type of a meta-search engine, and we utilized Google search engine (www.google.com). We chose Google since it is one of the best Internet search engines.

The system begins with a considerable number of results from a search engine. In this experiment, we used the 100 top-ranked pages. In some cases, the search engine provides less than 100 results. We also excluded broken Web pages.

The system gets an expanded word list associated with the query. The expansion is performed by the connection network (CN). The Web pages are evaluated using the expanded vocabulary. Each page is transformed to a vector with TF-IDF method [20] [8] before the evaluation. Finally, Web pages are ranked by the document evaluator which was tuned by GA.

3 Methodology

3.1 Connection Network

To date, a number of techniques were suggested for representing the relationship between words as a network. They were used for various fields such as natural language processing, Web search, etc. [15] [5] [7]. These techniques represent the conceptual relationship between words [15], connect the related words [5], or classify words under the categories [7]. However, these techniques considered only the existence of relationship and did not quantify the degree of relationship between words.

The problem of quantifying the relationship between words can be considered as a special case of the "market-basket" problem. The "market-basket" problem is a general problem to quantify the degree of relationship between items in a market. Various metrics have been introduced to quantify the relationship be-

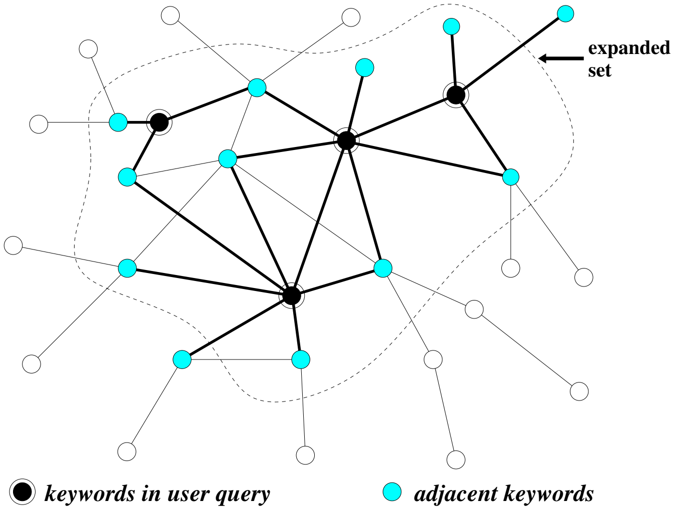

Figure 2: Semiotic connection network

tween items [2] [1] [3] [9] [6]. We design a new metric to quantify the relationship between words and construct a connection network to represent the relationship based on the metric.

3.1.1 Metric for the degree of relationship

Let D be the whole document set and W be the whole word set. For two words $x, y \in W$, we define $n(x)$, $n(y)$ to be the number of documents in which the words x and y occur, respectively, and define $n(x, y)$ to be the number of documents in which both x and y occur. We define the function $f : W \times W \longmapsto R$ to represent the degree of relationship between x and y as follows (R : the set of real numbers) :

$$f(x,y) = \begin{cases} \dfrac{log(n(x,y)) + C}{n(x) + n(y)} & \text{, if } \quad n(x,y) \neq 0 \\ 0 & \text{, if } \quad n(x,y) = 0 \end{cases}$$

, where C is a constant.

The function f has a similar shape to *interest* [3] or *similarity* [6] in data mining area.[1]

3.1.2 Keyword Expansion Based on Connection Network

We represent each word as a vertex. For each pair of words x and y such that $n(x, y) \neq 0$, we connect two vertices x and y with an edge and assign $f(x, y)$ as the weight of the edge. Then, we have the connection-network graph $G = (V, E)$ where V is the set of vertices and E is the set of edges.

Using the connection network, we process the expansion of keywords as follows. We assume that a user asked a query which consists of x_1, x_2, ... , x_k ($x_i \in W, 1 \leq i \leq k$) . Let $N(x_i)$ be a neighbor set of x_i ($1 \leq i \leq k$) on the connection network. We define the keyword expansion score function $s : W \longmapsto R$ as follows:

$$s(y) = \begin{cases} \displaystyle\sum_{1 \leq i \leq k} f(x_i, y) & \text{, if } \quad y \in \displaystyle\bigcup_i N(x_i) \\ 0 & \text{, otherwise .} \end{cases}$$

We choose a set of words with large enough expansion scores.

3.2 TF-IDF (Term Frequency, Inverse Document Frequency)

TF-IDF is a method to represent a document in a vector space [20] [8]. Each word in the document is assigned a scalar value. The scalar value reflects the relative importance of the word in the document and in the whole document set.

For a word w and a document d containing the word, TF(w, d) means the frequency that the word w occurs in the document d. DF(w) means the number of documents in which the word w occurs. IDF is defined as

$$IDF(w) = log \tfrac{|D|}{DF(w)}$$

where D is the set of all documents and $|D|$ is the

[1] $interest(x, y) = \frac{|D| \cdot n(x,y)}{n(x) \cdot n(y)}$ and $similarity(x, y) = \frac{n(x,y)}{n(x)+n(y)-n(x,y)}$.

number of all the documents.

A vector element $d^{(i)}$ associated with a word w_i is represented by the product of TF and IDF.

$$d^{(i)} = TF(w_i, d) \times IDF(w_i)$$

Then, the document d can be represented by a vector

$$\vec{d} = (d^{(1)}, d^{(2)}, ..., d^{(n)}).$$

From these scalar vectors we can compare the similarities of documents and evaluate a document. The similarity of two documents, V_j and V_k, is defined as

$$Sim(V_j, V_k) = \frac{V_j \cdot V_k}{|V_j| \times |V_k|} \qquad j, k \in \{1, 2, ..., n\}$$

where $V_j \cdot V_k$ means the inner product of V_j and V_k, and $|V_j|$ and $|V_k|$ mean the norms of V_j and V_k, respectively.

3.3 Document Evaluation

To find a document which a user wants the most, we should extract information as much as possible from queries, expanded-words obtained from the connection network, and other elements. Because the elements have different roles and relative importance, we need a process to optimize their roles. To optimize document evaluation, we should choose the documents that users have intended to find.

Let U be a set of words and t be a document. We define the TF-IDF vector related to U and t, $V_{U,t}$, as

$$\overrightarrow{V}_{U,t} = (v_1, v_2, ..., v_{|U|})$$

where $v_i = TF(w_i, t) \times IDF(w_i)$ and $w_i \in U$.

In Section 3.1.2, we showed that the connection network not only gives expanded words, but also gives the real values that represent the strengths between queries and the words. We denote by $S_{U,t}$ the score vector related to a word set U and a document t.

We define the vector $S_{U,t}$ as

$$\overrightarrow{S}_{U,t} = (s_1, s_2, ..., s_{|U|})$$

where

$$s_i = \begin{cases} s(w_i) & \text{, if } \quad w_i \text{ occurs in } t \text{ for } w_i \in U \\ 0 & \text{, otherwise .} \end{cases}$$

, and $s(w_i)$ is the keyword expansion score value of w_i in Section 3.1.2

We denote by $W(q)$ and $W(CN)$, the sets of the words in the user query q and the expanded-words by the connection network, respectively. Let the document length of d be $l(d)$.

The attractiveness $e(d)$ of a document d is defined as

$$e(d) = \frac{a_1|V_{W(q),d}|^{x_1} + a_2|V_{W(CN),d}|^{x_2} + a_3|S_{W(CN),d}|^{x_3}}{a_4 \cdot l(d)^{x_4}} \tag{1}$$

.

In the above formula (1), a long document length is a disadvantage.

3.4 The Evaluation of Evaluation Methods

A set of parameters in the formula (1) corresponds to an evaluation method of documents. The GA attempts to find an optimal evaluation method, i.e. an optimal set of parameters. When a parameter set is generated in GA, its fitness has to be evaluated. This is the "evaluation of evaluation methods."

Among the several methods, recall-precision (RP) is usually used as an information retrieval standard for various studies [19][10]. Recall means the returned ratio among all the appropriate documents. Precision means the returned ratio of appropriate documents among all the appropriate documents [10].

Here, we suggest two metrics to evaluate evaluation methods. Internet search engines usually provide a lot of documents in response to a user query. However, the most important would be those in the first page. The first page usually shows around 10 URL-links. We focus on the 10 top-ranked URLs in the training. This is a concept altered from the RP method. Summing up the points of 10 top-ranked links is available if each document was rated previously.

We assume a document set has a ranking order by the points of formula (1) for each query. Let i be the ranking number of a document and $p_{q,i}$ be the rating of i-th ranked documents for a query $q \in Q$ (Q : all the query set).

The first measure for the attractiveness of evaluation methods is as follows :

$$fitness_1 = \sum_{q \in Q} \sum_{i=1}^{10} p_{q,i} \tag{2}$$

Our second measure gives a weight to each of the 10-ranked. From the tenth to the first, we assign 1.1 to 2.0 as the weights.

When i means the rank of a document and $p_{q,i}$ is the rating for the document and a query $q \in Q$, the second

```
steady-state GA
    create initial population $P$;
    repeat {
        Choose two chromosomes $p_1, p_2$;
        offspring = crossover($p_1, p_2$)
        offspring = mutation(offspring)
        replace (offspring, $P$);
    } until (stopping condition)
    return the best solution;
```

Figure 3: Steady-State GA Framework

a_1	a_2	a_3	a_4	x_1	x_2	x_3	x_4
1.02	2.13	1.03	3.03	2.11	0.54	2.33	0.22

Figure 4: Problem encoding example

measure is defined as

$$fitness_2 = \sum_{q \in Q} \sum_{i=1}^{10} \frac{(21-i)p_{q,i}}{10} \qquad (3)$$

We apply the above two expressions (2) and (3) to maximize each of the total amounts.

4 Parameter Tuning by Genetic Algorithm

We use a steady-state GA. The template is given in Figure 3. Based on the formula (1) of Section 3.3, we use the GA to search for an attractive parameter set $S = \{a_1, a_2, a_3, a_4, x_1, x_2, x_3, x_4\}$. The problem is to find the best set S maximizing the fitnesses of formulas (2) or (3).

- **Problem Encoding and Crossover**
  In the problem, the parameters are all real numbers. Each solution is a set of 8 parameter values. In our GA, a chromosome is represented by an array with real numbers. Each element of the array is called a gene and we restrict the range of each gene to [0.01,4].

  In a variable set $S = \{a_1, a_2, a_3, a_4, x_1, x_2, x_3, x_4\}$, four parameters a_1, a_2, a_3, and a_4 are coefficients. We assumed that the ratio among them would not be over 1:400. Because x_1, x_2, x_3, x_4 are exponents, we limit them real numbers in [0,4]. Figure 4 shows an example chromosome.

  We use the arithmetic crossover operator [14, pp.104-5]. It creates a new offspring by assigning the average of the corresponding gene values in the parents for each gene. Figure 5 shows an example crossover operator. We should note that a GA with the arithmetic crossover is prone to premature converge. The diversity of solutions needs to be carefully controlled by mutation.

*Parent*1

a_1	a_2	a_3	a_4	x_1	x_2	x_3	x_4
1.20	2.30	1.03	2.00	2.00	0.50	2.33	0.22

*Parent*2

a_1	a_2	a_3	a_4	x_1	x_2	x_3	x_4
4.80	4.30	1.01	4.00	2.20	0.70	2.67	0.44

offspring

a_1	a_2	a_3	a_4	x_1	x_2	x_3	x_4
3.00	3.30	1.02	3.00	2.10	0.60	2.50	0.33

Figure 5: Arithmetic crossover example

- **Selection, Mutation and Replacement**
  We use the tournament selection to choose parents. A parent chromosome is selected as a result of competition among a member of randomly chosen individuals.

  The GA then perturbs the offspring by mutation operator. It replaces each gene with a random number in the proper range with the probability 1/32.

  We replace a chromosome having the worst fitness in the population with the offspring.

5 Experimental Results

5.1 Experimental plan

The experiment begins with results from a search engine. When we finished the preparation with 33 queries, we had on average 85 Web pages for each query. Table 1 shows the statistics.

The eventual performance of the system relies on the users' satisfaction. The prepared Web pages (2812 in total) were rated from 1 to 5 by 11 people. To avoid any prejudice, we shuffled the pages for the evaluators not to know about the rankings produced by the search engine. The most satisfactory Web pages earned 5, and the least satisfactory pages earned 1. Table 2 shows the distribution of the evaluated results. The average rate of the pages returned by Google was 2.48.

We divide the data set into three disjoint sets. We perform three symmetric experiments on the sets. In each experiment, we choose one of them in turn as the test set and perform training with the other two sets. This is a type of experimental design called k-fold cross validation [16].

Table 1: Statistic of Collected Data

Number of queries	33
Number of web pages	2812
Average number of web pages per query	85
Average rating	2.48

Table 2: Distribution of Ratings

Point	Number of Web-pages
1	1020
2	571
3	486
4	330
5	405
Total	2812

We used a steady-state GA with population size 50. If the same chromosomes are generated for five consecutive iterations, we assume the population has converged and stop the GA.

The training set is again divided into real-training set and validation set. The validation set is not directly used for parameter tuning but used for monitoring over-fitting. The performance on the training set usually shows a monotonic increase; on the other hand, that on the validation set usually shows a bitonic curve. We take the solution that corresponds to the peak of the bitonic curve. Finally, we test with the remaining test set and compare the result with the Google's search result.

5.2 Results

We set $k = 3$ for k-fold cross validation. Both of the two formulas of Section 3.4 were used for evaluation.

The Table 3 and Table 4 show the experimental results. Overall, the suggested system showed 6% improvement of satisfaction against Google's search results.

6 Conclusions and Future Work

In this paper we introduced a search ranking method that uses a semiotic connection network to retrieve contextual words.

From the experimental results, we can conclude that i) the connection network helps a search engine better satisfy the intentions of users, and ii) the GA tunes up parametric factors needed for document evaluation, and helps better ranking.

Table 3: 3-Fold Cross Validation with Formula 2

Test Set	1	2	3	Sum
Google	324	268	315	907
Our System	329	302	326	957
Performance	1.02	1.13	1.03	1.06(Avg)

Table 4: 3-Fold Cross Validation with Formula 3

Test Set	1	2	3	Sum
Google	508.4	411.4	493.7	1413.5
Our System	527.5	467.4	508.4	1503.3
Performance	1.04	1.14	1.03	1.06(Avg)

We should also note that the suggested method is not for a full search engine such as Google, Yahoo, etc; it can be used as an engine inside a full search engine or as a postprocessor for re-ranking the results. Currently it is incorporated into a commercial product.

The connection network is under reinforcement process with more document data. We expect the quality of ranking to be improved with a stronger connection network.

Acknowledgements

This work was partly supported by KOSEF through Statistical Research Center for Complex Systems at Seoul National University and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

[1] R. Agrawal, T. Imilienski, and A. Swami. Database mining: A performance perspective. *IEEE Trans. on Knowledge and Data Engineering*, 5(6):914–925, 1993.

[2] R. Agrawal, T. Imilienski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, pages 207–216, 1993.

[3] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, pages 255–264, 1997.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Net-*

work and ISDN Systems, 30(1-7):107–117, April 1998.

[5] L. Chen and K. Sycara. Webmate: A personal agent for browsing and searching. In *Autonomous Agents*, Minneapolis, May 1998.

[6] E. Cohen, M. Datar, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Trans. on Knowledge and Data Engineering*, 13(1):64–78, 2001.

[7] Semio Corp. *Connecting to Your Knowledge Nuggets.* Semio Corp. White Paper, 2001.

[8] D. Fragoudis and S. D. Likothanassis. Retriever: A self-training agent for intelligent information discovery. In *IEEE Symposium on Information and Intelligent Agents*, 1999.

[9] R. J. Bayardo Jr. and R. Agrawal. Mining the most interesting rules. In *Proc. of the fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 145–154, 1999.

[10] M. Junker, R. Hoch, and A. Dengel. On the evaluation of document analysis components by recall, precision, and accuracy. In *International Conference on Document Analysis and Recognition*, pages 713–16, Los Alamitos, CA, USA, 1999.

[11] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. S. Tomkins, and E. Upfal. The web as a graph. In *Symposium on Principles of Database Systems*, 2000.

[12] M. L. Mauldin. Lycos: Hunting WWW information. Technical report, CMU, 1994.

[13] O. A. McBryan. GENVL and WWWW: Tools for taming the web. In *First International Conference on the World Wide Web*, CERN, Geneva(Switzerland), 1994.

[14] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer, 1992.

[15] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *Int'l Journal of Lexicography*, 3(4):235–244, 1990.

[16] N.J. Nilsson. *Artificial Intelligence : A New Synthesis.* Morgan Kaufmann Publishers, Inc, 1998.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[18] B. Pinkerton. Finding what people want: Experiences with the WebCrawler. In *The Second International WWW Conference*, Chicago, USA, 1994.

[19] G. Salton. *Automatic Text Processing; the Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley Publishing Company, 1989.

[20] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–979, 1991.

[21] T. Taketa and H. Nukokawa. An efficient information retrieval method in WWW using genetic algorithms. In *Parallel Execution on Reconfigurable Hardware (PERH). IEEE.*, 1999.

[22] C.C. Yang, J. Yen, and H. Chen. Intelligent Internet searching engine based on hybrid simulated annealing. In *International Conference on System Sciences*, pages 415–22, Hawaii, USA, 1998.