# Fusion Approach for Optimizing Web Search Performance*

## 웹 검색 성능 최적화를 위한 융합적 방식

Kiduk Yang (양기덕)**


### ABSTRACT

This paper describes a Web search optimization study that investigates both static and dynamic tuning methods for optimizing system performance. We extended the conventional fusion approach by introducing the "dynamic tuning" process with which to optimize the fusion formula that combines the contributions of diverse sources of evidence on the Web. By engaging in iterative dynamic tuning process, where we successively fine-tuned the fusion parameters based on the cognitive analysis of immediate system feedback, we were able to significantly increase the retrieval performance.Our results show that exploiting the richness of Web search environment by combining multiple sources of evidence is an effective strategy.

### 초 록

이 논문은 시스템 성능을 최적화하기 위해 정적 및 동적 튜닝 방법을 이용한 웹 융합검색 연구의 내용을 보고합니다. 기존의 융합 방식을 넘어선 "다이나믹 튜닝"이라는 과정을 도입하여 웹의 다양한 정보소스의 기여를 최적화 시킬 수 있는 융합 공식을 생성하는 방법을 조사한 이 연구의 결과는 웹 검색 환경의 풍요로운 여러 데이터 소스를 활용하는 것이 효과적인 전략이라는 것을 보여주었습니다. 본 연구에서는 즉각적인 시스템 피드백 인지분석을 기반으로 융합 매개 변수를 미세 조정하는 반복적 인 다이나믹 튜닝 과정을 통해 크게 검색 성능을 향상시킬 수 있었습니다.


Keywords: fusion, information retrieval, Web search, performance optimization, dynamic tuning
융합, 정보검색, 웹 검색, 성능 최적화, 다이나믹 튜닝

---

* The study reported in the paper extends a prior study by Yang and Yu (2005).

** Kyungpook National University. Department of Library and Information Science. Daegu, Korea (kiyang@knu.ac.kr)

■ 논문접수일자 : 2014년 11월 28일 ■ 최초심사일자 : 2014년 11월 28일 ■ 게재확정일자 : 2015년 3월 9일
■ 정보관리학회지, 32(1), 7-22, 2015. 〔http://dx.doi.org/10.3743/KOSIM.2015.32.1.007〕

## 1. Introduction

Despite the apparent success of popular Web search engines such as Google and Naver, the problem of information retrieval on the Web is still far from being solved. While commercial search engines perform reasonably well in targeted tasks of known item search and simple search of specific information, satisfying more complex information needs that require comprehensive retrieval of relevant information at top ranks (i.e., high precision and high recall) is a problem for search industry as well as information retrieval (IR) research community.

As the size, diversity, and complexity of information on the Web grows astronomically, we should look beyond simple search to leverage the Web information in such a way to facilitate the understanding as well as discovery of information. To that end, we need to not only utilize multiple sources of evidence and integrate a variety of methodologies, but also combine human capabilities with those of the machine.

This paper describes our investigation to optimize Web search performance by leveraging human and computer capabilities while combining multiple sources of evidence and various IR methods. Specifically, we employed both static and dynamic tuning methods to optimize the fusion formula that combines multiple sources of evidence and methods. By static tuning, we refer to the typical stepwise tuning of system parameters based on training data. "Dynamic tuning", the key idea of which is to combine the human intelligence, especially pattern recognition ability, with the computational power of the machine, involves an interactive system tuning process that facilitates fine-tuning of the system parameters based on the cognitive analysis of immediate system feedback.

In order to investigate our fusion optimization approach for Web search, we implemented an experimental Web IR system in the WIDIT infrastructure[1)] and tested its performance with the TREC Web track data. The next section discusses related work in Web information retrieval research, section 3 details our methodology, followed by the description of the experiment in section 4 and the discussion of results in section 5.

## 2. Related Research

Web IR is riddled with challenges not encountered in the homogeneous and controlled environment of traditional IR research. The complexity and richness of the Web search environment call for approaches that extend conventional IR methods to leverage rich sources of information on the Web. Text Retrieval Conference (TREC), which is an international conference that investigates the efficacy of various IR approaches in a standardized setting, has been a fertile ground for cutting-edge IR research. In the Web IR experiment of TREC, otherwise known as the Web track, many TREC participants explored meth-

1) WIDIT (Web Information Discovery Integrated Tool, http://widit.knu.ac.kr/) is a research infrastructure constructed and maintained by the author.

ods of leveraging non-textual sources of information such as hyperlinks and document structure. The general consensus among the early Web track participants was that link analysis and other non-textual methods did not perform as well as the content-based retrieval methods fine-tuned over the years (Hawking et al., 1999; Hawking et al., 2000; Gurrin & Smeaton, 2001; Savoy & Rasolofo, 2001).

There have been many speculations as to why link analysis, which showed much promise in previous research and has been so readily embraced by commercial Web search engines, did not prove useful in Web track experiments. Most such speculations point to potential problems with Web track's earlier test collections, from the inadequate link structure of truncated Web data (Savoy & Picard, 1998; Singhal & Kazkiel, 2001), and relevance judgments that penalize the link analysis by not counting the hub pages as relevant (Voorhees & Harman, 2000) and reward the content analysis by counting multiple relevant pages from the same site as relevant (Singhal & Kazkiel, 2001), to unrealistic queries that are too detailed and specific to be representative of real world Web searches (Singhal & Kaszkiel, 2001).

In an effort to address the criticism and problems associated with the early Web track experiments, TREC replaced its earlier Web test collection of randomly selected Web pages with a larger and potentially higher quality domain-specific collection. Adjustment of the Web track environment brought forth renewed interest in retrieval approaches that leverage Web-specific sources of evidences such as link structure and document structure.

For the task of finding the entry page of a specific site described by the query (i.e. homepage finding task), Web page's URL characteristics, such as its type and length, as well as the anchor text of Web page's inlinks proved to be useful sources of information to be leveraged (Hawking & Craswell, 2002).

In the topic distillation task, which requires finding a short, comprehensive list of pages that are good information resources, anchor text seemed to be a useful resource, especially as a mean to boost the performance of content-based methods via fusion (e.g., result merging) (Hawking & Craswell, 2002; Craswell & Hawking, 2003). Various site compression strategies, which attempt to select the "best" pages of a given site, was another common theme in the topic distillation task, once again demonstrating the importance of fine-tuning the retrieval system according to the task at hand (Amitay et al., 2003; Zhang et al., 2003). It is interesting to note that link analysis (e.g. PageRank, HITS variations) did not prove itself to be an effective strategy and the content-based method seems to be still the most dominant factor in the Web track. In fact, the two best results in topic distillation task were achieved by the baseline systems that used only the content-based methods (MacFarlane, 2003; Zhang et al., 2003).

In our earlier studies (Yang, 2002a, 2002b), where we investigated various fusion approaches, we found that simplistic approach combining the results of content- and link-based retrieval results did not enhance retrieval performance in general. Our study is motivated by the belief that retrieval performance

of static fusion approach such as weighted result merging can be enhanced via a more dynamic approach to fusion.

## 3. Methodology

Based on the assumption that the key to effective Web IR lies in exploiting the richness of Web search environment by combining multiple sources of evidence, we focused our efforts on extending and optimizing the fusion methods. Our approach to combining multiple sources of evidence is twofold. First, we combine multiple sets of retrieval results generated from multiple sources of evidence (e.g. body text, anchor text, header text) and multiple query formulations using a weighted sum formula, whose parameters are tuned via a static tuning process using training data (Bartell et al., 1994; Modha and Spangler, 2000; Yang, 2014). The ranking of the static fusion result is then "optimized" via a dynamic tuning process that involves iterative refining of fusion formula that combines the contributions of diverse Web-based evidence (e.g. hyperlinks, URL, document structure). The dynamic tuning process is implemented as a Web application; where interactive system parameter tuning by the user produces in real time the display of system performance changes as well as the new search results annotated with metadata of fusion parameter values (e.g. link counts, URL type, etc.). The key idea of dynamic tuning, which is to combine the human intelligence, especially pattern recognition ability, with the computational power of the machine, is implemented in this Web application that allows human to examine not only the immediate effect of his/her system tuning but also the possible explanation of the tuning effect in the form of data patterns.

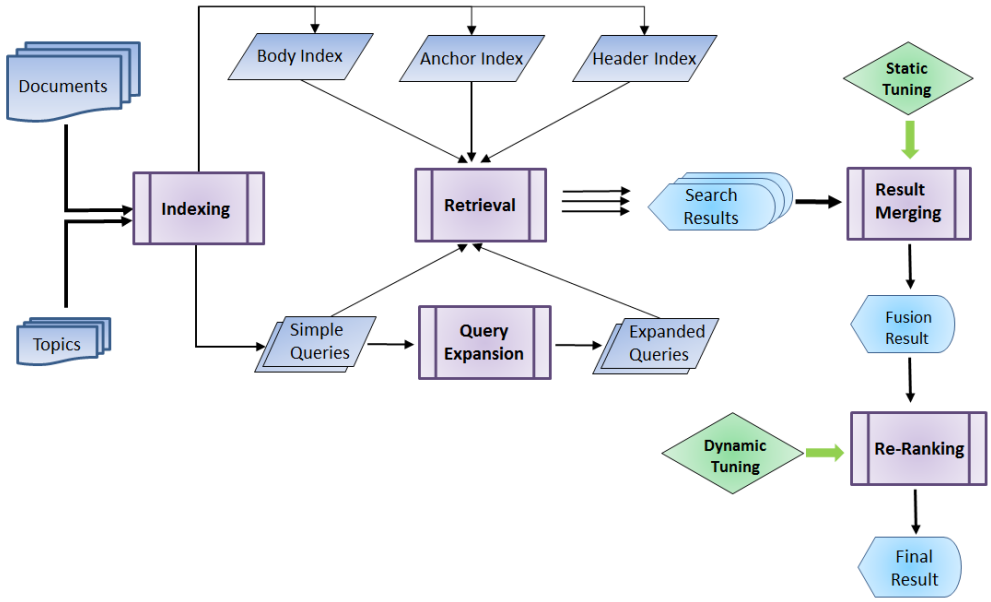
### 3.1 System Overview

Our experimental Web IR system consists of five main components: indexing, query expansion, retrieval, fusion (i.e. result merging), and reranking modules.

The indexing module processes various sources of evidence to generate multiple indexes. The query expansion module adds related terms to the original query by mining data sources on the Web. The retrieval module produces multiple result sets from using different query formulations against multiple indexes. The fusion module, which is optimized via the static tuning process, combines result sets using a weighted sum formula. The reranking module uses query type-specific reranking formulas optimized via dynamic tuning process to rerank the merged results. Figure 1 shows an overview of the system architecture.

### 3.2 Indexing

The indexing module preprocesses Web documents by removing HTML tags and stopwords and applying the simple plural remover (Frakes & Baeza-Yates, 1992). The stopwords consist of non-meaningful words such as words in a standard stopword list, non-alphabetical words, words consisting of more than 25




〈Figure 1〉 Web IR System Architecture

or less than 3 characters, and words that contain 3 or more repeated characters. Hyphenated words are split into parts before applying the stopword exclusion, and acronyms and abbreviations are kept as index terms.

In addition to extracting body text terms (i.e. terms between <body> and </body> tags) to create a *body index* (inverted index consisting of terms from body text), we extract terms from document title, meta keywords and descriptions, and "emphasized" text (e.g. text with <b>, , <font>, <u>, <h1> tags) to create a *header index*, and extract terms from the anchor texts of incoming links to create an anchor *index*. Thus, the indexing module creates three sets of term indexes: first based on document content (*body index*), second based on document structure (*header index*), and third based on link structure (*anchor index*).

### 3.3 Query Expansion

The main objective of query expansion is to add related terms to the original query so as to make the query more descriptive. Among the common query expansion strategies of pseudo-feedback, syntactic expansion by thesaurus (e.g., WordNet), and Web-based expansion, we chose the Web-based expansion to strengthen the queries. Web queries tend to be one or two word descriptions of target entities and thus do not benefit much by syntactic expansion (e.g., synonym expansion). Pseudo-feedback, which relies on top ranked documents being relevant, can be problematic for short queries that produce poor initial retrieval results due to incomplete descriptions of entities. Web-based expansion, on the other hand, searches much larger external data sources of the

Web, and has shown to be an effective query expansion strategy for difficult queries (Kwok, Grunfeld, & Deng, 2005).

Our Web-based query expansion (QE) consists of the Wikipedia QE module, which extracts terms from Wikipedia articles and Wikipedia Thesaurus, and the Google QE module, which extends the PIRC approach that harvests expansion terms from Google search results (Kwok, Grunfeld, & Deng, 2005).

#### 3.3.1 Wikipedia QE (WQE) Module

WQE module generates two types of wiki-expanded queries. The first wiki-expansion starts by querying the Wikipedia search engine with the entire string of the short query. If the result is an encyclopedia article page, the title and top portion of the article up to the content listing is harvested for term selection. If the result is a full-text search result or disambiguation page that contains a list of potential matches, titles and snippets of the list items are harvested for further processing. The k most frequently occurring terms in the harvested text are added to the original query to create an expanded query.

The second type of wiki-expanded query consists of Wikipedia title and thesaurus terms. First, n-grams of decreasing length are extracted from the original query by a sliding window (e.g., “computer monitor price”, “computer monitor”, “monitor price”, “computer”, “monitor”, “price”) and checked against Wikipedia to identify phrases. The titles of Wikipedia pages (“e.g., “visual display unit”, “price”) retrieved by the longest n-grams (e.g., “computer monitor”, “price”) are then used to find synonyms and related terms from the Wikipedea Thesaurus[2]. Title terms from article pages are given the weight of 1, while title terms from search and disambiguation pages are given reduced weights (e.g., 0.8). The term weights of synonyms are even reduced further (1/2 of title term weights for synonyms, 1/3 for related terms). If the Wikipedia result is not an article, only the synonyms are added to reduce the adverse effect of incorrect expansion.

#### 3.3.2 Google QE (GQE) Module

While WQE mines the manually constructed knowledge base of Wikipedia, GQE utilizes the rich information on the Web effectively searched by Google to identify related terms. Like WQE, GQE module generates multiple types of google-expanded queries by varying the source and weighting of expansion terms. The first step of GQE is to query Google with the short query and harvest the titles, snippets, and full-texts of the top $n$ search results that are HTML pages. The first type of google-expanded query consists of top $k$ most frequently occurring terms from titles and snippets. The second and third types of google-expanded queries are composed of top $k$ weighted terms from the full-texts of search results, where the term weight is computed by a modified version of the local context analysis (LCA) formula (equation 2) using the original query and a combination of expanded queries.

The original LCA formula, shown in equation 1, selects expansion terms based on co-occurrence

2) http://dev.sigwp.org/WikipediaThesaurusV3/

with query terms, $co(c,w_i)$, and their frequency in the whole collection, $idf(c)$, normalized over $n$ (Xu & Croft, 2000). The *idf* component of LCA, which modifies the standard inverse document frequency with an upper bound, estimates the absolute importance of a term by its discriminating value, while the co-occurrence component estimates the relative term importance with respect to a given query. Since the collection frequency is unknown in the Web setting, we use term frequencies normalized by term distance to compensate for the lack of *idf*. The normalized frequency of term $c$ in document $d$, is computed by first summing the word distances between occurrences of $c$ and nearest query term $w_i$ in d and taking the inverse of its log value. The normalized frequency of query term $w_i$ in $d$ is computed in a similar manner by taking the inverse log of the sum of minimum word distances between occurrences of query term $w_i$ and $c$ in $d$. The normalized term frequency modifies the weight of each term occurrence with co-occurrence distance in order to reward terms that occur closer to query terms.

$$\begin{aligned} co\_\deg ree(c,w_i) &= \log_{10}(co(c,w_i)+1)idf(c)/\log_{10}(n) \\ co(c,w_i) &= \sum_{d\ in\ S} tf(c,d)tf(w_i,d) \\ idf(c) &= \min(1.0, \log_{10}(N/N_c)/5.0) \end{aligned} \quad (1)$$

$$\begin{aligned} co\_\deg ree(c,w_i) &= \log_{10}(co(c,w_i)+1)/\log_{10}(n) \\ co(c,w_i) &= \sum_{d\ in\ S} tf_{norm}(c,d)tf_{norm}(w_i,d) \\ tf_{norm}(c,d) &= \sum_{c\ in\ d} \frac{1}{\log_5(\min dist(c,w_i)+1)} \\ tf_{norm}(w_i,d) &= \sum_{w_i\ in\ d} \frac{1}{\log_5(\min dist(w_i,c)+1)} \end{aligned} \quad (2)$$

### 3.3.3 Query Fusion

After generating expanded queries, we produced combined QE queries by selecting terms from different query expansion types. For term selection and weighting, we devised ad-hoc heuristics based on observation, trial and error, and some basic assumptions regarding the quality of QE types. A generalized form of QE query fusion heuristic is described below.

1. Merge top m terms from each expanded query.
2. Compute fusion term weights (twf).
   a. if merged from wiki-QE and google-LCA, twf = 10/rank
   b. else if from wiki-QE, twf = 5/rank
   c. else if google-LCA, twf = 3/rank
   d. else twf = 1/rank
3. Select top n unigrams and top n bigrams by fusion term weight.

## 3.4 Retrieval Module

The retrieval module implements both Vector Space Model (VSM) using the SMART length-normalized term weights and the probabilistic model using the Okapi BM25 formula. Documents are ranked in decreasing order of the inner product of document and query vectors,

$$\mathbf{q}^{\mathrm{T}}\mathbf{d}_i = \sum_{k=1}^{t} q_k d_{ik} \quad (3)$$

where $q_k$ is the weight of term $k$ in the query, $d_{ik}$ is the weight of term $k$ in document $i$, and $t$ is the number of terms in the index.

For the VSM implementation, SMART *Lnu* weights

with the slope of 0.3 are used for document terms (Buckley et al., 1997), and SMART *ltc* weights (Buckley et al., 1995) are used for query terms. *Lnu* weights attempt to match the probability of retrieval given a document length with the probability of relevance given that length (Singhal, Buckley, & Mitra, 1996).

$$d_{ik} = \frac{\log(f_{ik})+1}{\sqrt{\sum_{j=1}^{t}(\log(f_{ij})+1)^2}}$$
$$q_k = \frac{(\log(f_k)+1)*idf_k}{\sqrt{\sum_{j=1}^{t}[(\log(f_j)+1)*idf_j]^2}} \quad (4)$$

Equation (4) describes the SMART formula, where $d_{ik}$ is the document term weight (*Lnu*), $q_k$ is the query term weight (*ltc*), $f_{ik}$ is the number of times term $k$ appears in document $i$, $f_k$ is the number of times term $k$ appears in the query, $idf_k$ is the inverse document frequency of term $k$, and $t$ is the number of terms in document or query.

The simplified version of the Okapi BM25 relevance scoring formula (Robertson & Walker, 1994), which is used to implement the probabilistic model, is described in equation (5), where $N$ is the number of documents in the collection, *df* is the document frequency, *dl* is the document length, *avdl* is the average document length, and k1, b, k3 are parameters (1.2, 0.75, 7 to 1000, respectively).

$$d_{ik} = \log\left(\frac{N-df_k+0.5}{df_k+0.5}\right)\frac{f_{ik}}{k_1((1-b)+b\cdot(\frac{dl}{avdl}))+f_{ik}}$$
$$q_k = \frac{(k_3+1)f_k}{k_3+f_k} \quad (5)$$

## 3.5 Fusion Module

The fusion module combines the multiple sets of search results after retrieval time. In addition to two of the most common fusion formulas, *Similarity Merge* (Fox & Shaw, 1995; Lee, 1997) and *Weighted Sum* (Bartell et al., 1994; Thompson, 1990), the fusion module employs variations of the weighted sum formula. The similarity merge formula multiplies the sum of fusion component scores for a document by the number of fusion components that retrieved the document (i.e. overlap), based on the assumption that documents with higher overlap are more likely to be relevant. Instead of relying on overlap, the weighted sum formula sums fusion component scores weighted with the relative contributions of the fusion components that retrieved them, which is typically estimated based on training data. Both formulas compute the fusion score of a document by a linear combination of fusion component scores.

In our earlier study (Yang, 2002b, 2014), similarity merge approach proved ineffective when combining content- and link-based results, so we devised three variations of the weighted sum fusion formula, which were shown to be more effective in combining fusion components that are dissimilar (Yang, 2002a). Equation (6) describes the simple *Weight Sum* (WS) formula, which sums the normalized system scores multiplied by system contribution weights. Equation (7) describes the *Overlap Weight Sum* (OWS) formula, which multiplies the WS score by overlap. Equation (8) describes the *Weighted Overlap Weighted Sum* (WOWS) formula, which multiplies

the WS score by overlap weighted by system contributions:

$$FS_{WS} = \sum(w_i * NS_i) \quad (6)$$

$$FS_{OWS} = \sum(w_i * NS_i * olp) \quad (7)$$

$$FS_{WOWS} = \sum(w_i * NS_i * w_i * olp) \quad (8)$$

where:
$FS$ = fusion score of a document,
$w_i$ = weight of system $i$,
$NS_i$ = normalized score of a document by system $i$,
$= (S_i - S_{min}) / (S_{max} - S_{min})$
$olp$ = number of systems that retrieved a given document.

The normalized document score, $NS_i$, is computed by Lee's min-max formula (1997), where $S_i$ is the retrieval score of a given document and $S_{max}$ and $S_{min}$ are the maximum and minimum document scores by method $i$.

One of the main challenges in using the weighted fusion formula lies in determination of the optimum weights for each system (*wi*). In order to optimize the fusion weights, we employ a static tuning process, where various weight combinations (e.g. 0.9 for body text, 0.08 for header text, 0.02 for anchor text) are evaluated with the training data of past TREC Web track results in a stepwise fashion.

## 3.6 Reranking Module

In order to optimize retrieval performance in top ranks, fusion results are reranked based on the content- and link-based evidences (e.g. hyperlinks, URL, document structure). The reranking heuristic consists of a set of ranking and document score boosting rules arrived at by dynamic tuning process involving interactive retrieval and manual system tuning in real time. The dynamic tuning process is applied to the best single and best fusion systems to "tune" the ranking heuristic.

The dynamic tuning component produces retrieval results that display individual scores for each source of evidence such as inter/intrasite in/outdegree, phrase/proximity match counts in body/header/anchor texts, and query term matches in URL as well as ranking and retrieval scores before/after the adjustment of reranking parameters by dynamic tuning.

### 3.6.1 Reranking Factors

TREC participants found various sources of evidence such as anchor text (Craswell, Hawking, & Robertson, 2001; Hawking & Craswell, 2002; Craswell & Hawking, 2003) and URL characteristics (Kraajj et al., 2002; Tomlinson, 2003, Zhang et al., 2003) to be useful in the Web track tasks. Based on those findings as well as the analysis of our previous Web IR studies, we decided to focus on four categories of the reranking factors. The first category is the field-specific match, where we score each document by counting the occurrences of query words (keyword, acronym, phrase) in URL, title, header, and anchor texts. The second category of reranking factors we use is the exact match, where we look for exact match of query text in title, header, and anchor texts (exact), or in the body text (exact2)

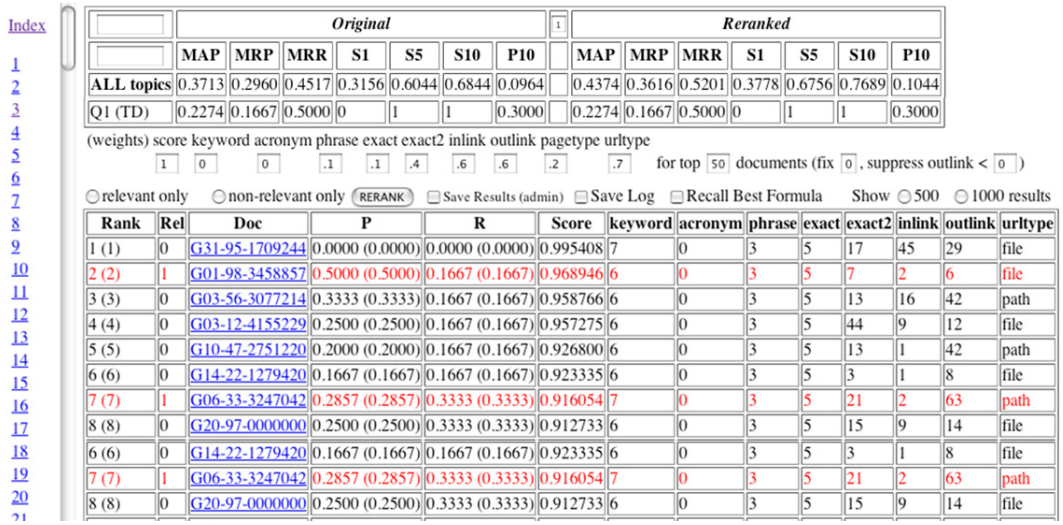
of documents. The third category is link-based, where we count documents' inlinks (indegree) and outlinks (outdegree). The last category is the document type, which is derived based on its URL (Tomlinson, 2003; Kraajj et al., 2002).

### 3.6.2 Dynamic Tuning

The dynamic tuning interface is implemented as a Web application (Figure 2); where interactive system parameter tuning by the user produces in real time the display of system performance changes as well as the new search results annotated with metadata of fusion parameter values (e.g. link counts, URL type, etc.).

The key idea of dynamic tuning, which is to combine the human intelligence, especially pattern recognition ability, with the computational power of the machine, is implemented in this Web application that allows human to examine not only the immediate effect of his/her system tuning but also the possible explanation of the tuning effect in the form of data patterns. By engaging in iterative dynamic tuning process that successively fine-tune the fusion parameters based on the cognitive analysis of immediate system feedback, we can increase system performance without resorting to an exhaustive evaluation of parameter combinations, which can not only be prohibitively resource intensive with numerous parameters but also fail to produce the optimal outcome due to its linear approach to fusion components combination.

The dynamic tuning interface, as can be seen in Figure 2, has a navigation pane on the left with query numbers, a click of which will populate the main display pane on the right. The main display pane has three horizontal components: side-by-side performance scores for original and reranked retrieval results at the top, weight specification form


Index

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

| | Original | | | | | | | | Reranked | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAP | MRP | MRR | S1 | S5 | S10 | P10 | | MAP | MRP | MRR | S1 | S5 | S10 | P10 |
| ALL topics | 0.3713 | 0.2960 | 0.4517 | 0.3156 | 0.6044 | 0.6844 | 0.0964 | | 0.4374 | 0.3616 | 0.5201 | 0.3778 | 0.6756 | 0.7689 | 0.1044 |
| Q1 (TD) | 0.2274 | 0.1667 | 0.5000 | 0 | 1 | 1 | 0.3000 | | 0.2274 | 0.1667 | 0.5000 | 0 | 1 | 1 | 0.3000 |

(weights) score keyword acronym phrase exact exact2 inlink outlink pagetype urltype

1 0 0 .1 .1 .4 .6 .6 .2 .7 for top 50 documents (fix 0 , suppress outlink < 0 )

relevant only non-relevant only RERANK Save Results (admin) Save Log Recall Best Formula Show 500 1000 results

| Rank | Rel | Doc | P | R | Score | keyword | acronym | phrase | exact | exact2 | inlink | outlink | urltype |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 (1) | 0 | G31-95-1709244 | 0.0000 (0.0000) | 0.0000 (0.0000) | 0.995408 | 7 | 0 | 3 | 5 | 17 | 45 | 29 | file |
| 2 (2) | 1 | G01-98-3458857 | 0.5000 (0.5000) | 0.1667 (0.1667) | 0.968946 | 6 | 0 | 3 | 5 | 7 | 2 | 6 | file |
| 3 (3) | 0 | G03-56-3077214 | 0.3333 (0.3333) | 0.1667 (0.1667) | 0.958766 | 6 | 0 | 3 | 5 | 13 | 16 | 42 | path |
| 4 (4) | 0 | G03-12-4155229 | 0.2500 (0.2500) | 0.1667 (0.1667) | 0.957275 | 6 | 0 | 3 | 5 | 44 | 9 | 12 | file |
| 5 (5) | 0 | G10-47-2751220 | 0.2000 (0.2000) | 0.1667 (0.1667) | 0.926800 | 6 | 0 | 3 | 5 | 13 | 1 | 42 | path |
| 6 (6) | 0 | G14-22-1279420 | 0.1667 (0.1667) | 0.1667 (0.1667) | 0.923335 | 6 | 0 | 3 | 5 | 3 | 1 | 8 | file |
| 7 (7) | 1 | G06-33-3247042 | 0.2857 (0.2857) | 0.3333 (0.3333) | 0.916054 | 7 | 0 | 3 | 5 | 21 | 2 | 63 | path |
| 8 (8) | 0 | G20-97-0000000 | 0.2500 (0.2500) | 0.3333 (0.3333) | 0.912733 | 6 | 0 | 3 | 5 | 15 | 9 | 14 | file |
| 6 (6) | 0 | G14-22-1279420 | 0.1667 (0.1667) | 0.1667 (0.1667) | 0.923335 | 6 | 0 | 3 | 5 | 3 | 1 | 8 | file |
| 7 (7) | 1 | G06-33-3247042 | 0.2857 (0.2857) | 0.3333 (0.3333) | 0.916054 | 7 | 0 | 3 | 5 | 21 | 2 | 63 | path |
| 8 (8) | 0 | G20-97-0000000 | 0.2500 (0.2500) | 0.3333 (0.3333) | 0.912733 | 6 | 0 | 3 | 5 | 15 | 9 | 14 | file |

〈Figure 2〉 Dynamic Tuning Interface

for fusion components (i.e. reranking factors) in the middle, and the ranked list of retrieved documents with individual fusion component scores at the bottom. The main idea is to discover patterns in fusion component scores across ranks that can be leveraged into improving retrieval performance by fine-tuning the fusion formula in the middle (i.e. reranking function).

The translation of discerned pattern into an effective weighting function is a trial-and-error process guided by a real-time display of performance gain or loss affected by the tuning. Sometimes the cognitive analysis of identified patterns suggests reranking heuristic that goes beyond a simple linear combination of reranking factors (e.g. rerank only top *n* results, with top *m* ranked fixed). In such cases, one must update the fusion formula component of the main display pane to accommodate the devised reranking heuristic. The dynamic tuning process as a whole is iterative because ew patterns emerge with each refinement of the fusion formula until the performance stabilizes.

# 4. Experiment

In order to evaluate the effectiveness of the our approach to Web IR, we conducted a series of retrieval experiments using the TREC's .GOV test collection, which consists of 1.25 million Web pages (18 GB) in .gov domain and 225 queries of mixed type (75 TD, 75 HP, 75 NP) and associated relevance judgments. For our study, we used the topic distillation (TD) queries only.

## 4.1 Initial Retrieval

As described in the methodology section, we created separate document indexes for body text, anchor text and header text and applied query expansion to construct various query formulations. Multiple queries against multiple indexes generated numerous retrieval sets for a given search in the initial retrieval phase.

## 4.2 Retrieval Optimization

The merging of the retrieval results were optimized via a static tuning process, where search results were combined using weighted sum with various weights. Optimizing the results of initial topic search is an efficient way to incorporate clues such as phrases and exact match (see section 3.6.1).

After the fusion optimization by static tuning, we employed a post-retrieval rank-boosting strategy to rerank the merged results for each query type using the dynamic tuning process. In order to assess the effectiveness of dynamic tuning, we devised a static reranking approach based on previous TREC research. Our static approach to reranking was as follows: boost the rank of potential homepages (identified by URL type determination) and pages with keyword matches in document titles and URLs while keeping top 5 ranks static.

We performed a series of dynamic tuning sessions using past TREC data, which involved repeated cycles of retrieval and tuning the reranking heuristic based on real time evaluation of retrieval results.

In contrast to static tuning, dynamic tuning process allows tuning of systems with numerous parameters by leveraging human intelligence. The main components of reranking heuristic we used were outdegree (e.g. boost score if large outdegree), phrase/proximity match (e.g. boost ranking if phrase match in title or anchor text), and query term match in URL (e.g. boost to top 10 rank if acronym match in URL).

# 5. Results

## 5.1 Query Expansion

Among various query expansion strategies, the Google-based expansion using the modified LCA weight with original query (equation 2) produced the best results. It even outperformed the query fusion results where expanded queries by different QE methods were combined. On the other hand, post-retrieval fusion (i.e., result merging) that combined the best results from QE groups did improve the results (Table 1).

For each QE group, there are many QE formulations depending on the number of search results used and number of expansion terms. Figure 3 plots the mean average precision (MAP) scores of gg2 queries[3] with varying number of expansion terms (3, 5, 10, …, 100) and figure 4 plots MAP with varying number of search results (3, 5, 10, 20, 30) from which to extract the terms.

The positive slopes in figure 3 indicate that more expansion terms are better for the Web-based query expansion. Jagged lines in figure 4, on the other hand, tell a slight different story. Each line in figure 4 represents queries with fixed term count. Without the uppermost line, which is the performance by queries of 110 terms, one may conclude that document count of 10 is optimal. In reality, however, 30-document queries outperform 10-document queries as the query becomes longer.

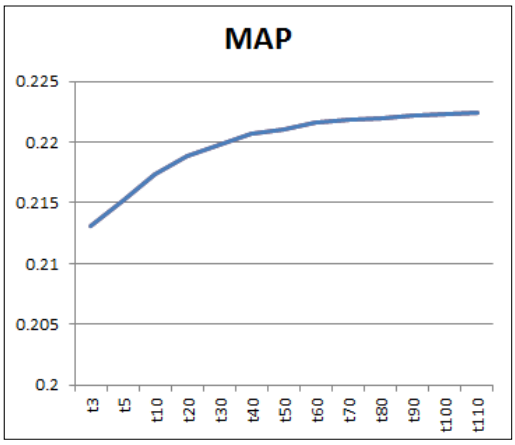
The results demonstrate that Web-based query expansion is an effective strategy for improving the performance of short queries (31% improvement over baseline). The marginal performance improvement by query fusion run suggests that QE term selection heuristics[4] should be optimized.

〈Table 1〉 Comparison of QE Methods

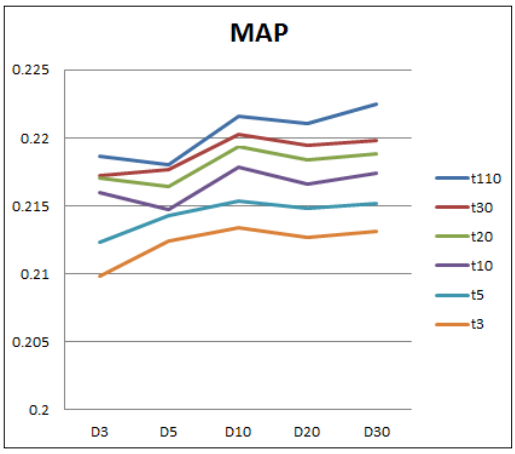
| | MAP |
|---|---|
| bestf | 0.2324 |
| gg2 | 0.2216 |
| wg | 0.2198 |
| gg | 0.2162 |
| gg3 | 0.2151 |
| wk | 0.2107 |
| s0 | 0.1694 |

bestf = fusion of best QE runs
gg = Google QE: title & snippets
gg2 = Google QE: LCA with original query)
gg3 = Google QE (LCA with combined query)
s0 = baseline query
wg = wk + gg + gg2 + gg3
wk = Wikipedia QE

3) gg2 queries in figure 6 have document count (i.e. number of search results used) of 30.
4) For combining the QE results, we did not optimize the fusion formula and simply used the fusion weights of 1's.




〈Figure 3〉 QE Performance by Term Count




〈Figure 4〉 QE Performance by Document Count

## 5.2 Reranking

Table 2 shows the performance improvement of fusion and reranking results over baseline, which is the best performing individual run. Fusion improved the baseline performance by 37%, static reranking improved the best fusion result by 8% (48% over baseline), and dynamic reranking run improved the static reranking result by another 16% (72% over baseline). It is clear from the table that both static and dynamic tuning for post-retrieval reranking are effective system performance optimization methods for leveraging diverse sources of evidence in Web IR.

〈Table 2〉 Fusion and Reranking Effect

| | MAP | $\Delta_{Baseline}$ | $\Delta$ |
|---|---|---|---|
| Baseline | 0.1694 | | |
| Best Fusion | 0.2324 | 37% | |
| Static Reranking | 0.2513 | 48% | 8% |
| Dynamic Reranking | 0.2918 | 72% | 16% |

The objective of reranking is to float low ranking relevant documents to the top ranks based on the post-retrieval analysis of reranking factors. Although reranking does not retrieve any new relevant documents (i.e. no recall improvement), it can produce high precision improvement via post-retrieval compensation (e.g. phrase matching). The key questions for reranking are what reranking factors to consider and how to combine individual reranking factors to optimize the reranking effect.

The effective reranking factors observed from the iterations of dynamic reranking were acronym, URLtype, and outdegree. In addition to harnessing both the human intelligence and machine processing power to facilitate the process of system tuning with many parameters, dynamic tuning turned out to be a good tool for failure analysis. We examined severe search failure instances via using the dynamic tuning interface and observed the following:

• Acronym Effect:

Documents about the acronym were ranked higher than those about the target topic. For instance, CDC homepage was ranked higher than documents about rabies for the query “CDC Rabies homepage”.

• Link Noise Effect:

Non-relevant documents with irrelevant links were ranked high. For example, relevant document for the query “Vietnam War” is Johnson Administration’s “Foreign Relations” document with 4 links to government documents about Vietnam, but our system retrieved pages about Vietnam with many irrelevant (e.g. navigational) links at top ranks.

• Topic Drift:

Topically related documents with high frequency of query terms were ranked high by WIDIT. For example, documents about drunk driving victims, MADD, etc. were ranked higher than the impaired driving program of NHTSA page for the “Drunk driving” query.

## 6. Discussion

We leveraged the richness of Web search environment by combining multiple sources of evidence and extended the conventional fusion approach by introducing the “dynamic tuning” process with which to optimize the contributions from multiple sources. By combining diverse sources of evidence on the Web and engaging in an iterative dynamic tuning process, where fusion parameters are successively fine-tuned via cognitive analysis of immediate system feedback, we were able to significantly enhance the retrieval performance and show that the fusion, especially with dynamic tuning, is a promising area of investigation for optimizing web search performance.

# References

Amitay, E., Carmel, D., Darlow, A., Lempel, R., & Soffer, A. (2003). Topic distillation with knowledge agents. Proceedings of the11th Text Retrieval Conference, 263-272.

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1994). Automatic combination of multiple ranked retrieval systems. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval.

Buckley, C., Salton, G., & Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. Proceeding of the 3rd Text Rerieval Conference, 1-19.

Buckley, C., Singhal, A., & Mitra, M. (1997). Using query zoning and correlation within SMART: TREC

5. Proceeding of the 5th Text REtrieval Conference, 105-118.

Craswell, N., & Hawking, D. (2003). Overview of the TREC-2002 Web track. Proceedings of the 11th Text Retrieval Conference, 86-95.

Craswell, N., Hawking, D., & Robertson, S. (2001). Effective site finding using link anchor information. Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval, 250-257.

Fox, E. A., & Shaw, J. A. (1995). Combination of multiple searches. Proceeding of the 3rd Text Rerieval Conference, 105-108.

Frakes, W. B., & Baeza-Yates, R. (Eds.). (1992). Information retrieval: Data structures & algorithms. Englewood Cliffs, NJ: Prentice Hall.

Gurrin, C., & Smeaton, A. F. (2001). Dublin City University experiments in connectivity analysis for TREC-9. Proceedings of the 9th Text Retrieval Conference, 179-188.

Hawking, D., & Craswell, N. (2002). Overview of the TREC-2001 Web track. Proceedings of the 10th Text Retrieval Conference, 25-31

Hawking, D., & Craswell, N., Thistlewaite, P., & Harman, D. (1999). Results and challenges in web search evaluation. Proceedings of the 8th WWW Conference, 243-252.

Hawking, D., Voorhees, E., Craswell, N., & Bailey, P. (2000). Overview of the TREC-8 web track. Proceedings of the 8th Text Retrieval Conference, 131-148.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. Proceedings of the 25th ACM SIGIR Conference on Research and Development in Information Retrieval, 27-34.

Kwok, K. L., Grunfeld, L., & Deng, P. (2005). Improving weak ad-hoc retrieval by Web assistance and data fusion. Information Retrieval Technology, 17-30. Springer Berlin Heidelberg.

Lee, J. H. (1997). Analyses of multiple evidence combination. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 267-276.

MacFarlane, A. (2003). Pliers at TREC 2002. Proceedings of the 11th Text Retrieval Conference, 152-155.

Modha, D., & Spangler, W. S. (2000). Clustering hypertext with applications to Web searching. Proceedings of the 11th ACM Hypertext Conference, 143-152

Robertson, S. E., & Walker, S. (1994). Some simple approximations to the 2-poisson model for probabilistic weighted retrieval. Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval, 232-241

Savoy, J., & Picard, J. (1998). Report on the TREC-8 experiment: Searching on the web and in distributed collections. Proceedings of the 8th Text Retrieval Conference, 229-240.

Savoy, J., & Rasolofo, Y. (2001). Report on the TREC-9 experiment: Link-based retrieval and distributed collections. Proceedings of the 9th Text Retrieval Conference, 579-516.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 21-29.

Singhal, A., & Kaszkiel, M. (2001). A case study in web search using TREC algorithms. Proceedings of the 11th International WWW Conference, 708-716.

Thompson. P. (1990). A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model. Information Processing & Management, 26(3), 371-382.

Tomlinson, S. (2003). Robust, web and genomic retrieval with hummingbird searchServer at TREC 2003. Proceedings of the 12th Text Retrieval Conference, 254-267.

Voorhees, E., & Harman, D. (2000). Overview of the eighth text retrieval conference. Proceedings of the 8th Text Retrieval Conference, 1-24.

Xu, J., & Croft, W.B. (2000). Improving the effectiveness of information retrieval with local context analysis. ACM Transaction on Information Systems, 18(1), 79-112.

Yang, K. (2002a). Combining text-, link-, and classification-based retrieval methods to enhance information discovery on the Web. (Doctoral Dissertation. University of North Carolina).

Yang, K. (2002b). Combining text- and link-based retrieval methods for web IR. Proceedings of the 10th Text Rerieval Conference, 609-618.

Yang, K. (2014). Combining multiple sources of evidence to enhance Web search performance. Journal of Korean Library and Information Science Society, 45(3), 5-36.

Yang, K., & N. Yu. (2005). WIDIT: Fusion-based approach to web search optimization. In Information Retrieval Technology, 206-220. Springer Berlin Heidelberg.

Zhang, M., Song, R., Lin, C., Ma, S., Jiang, Z., Jin, Y., Liu, Y., & Zhao, L. (2003). THU TREC 2002: Web Track Experiments. Proceedings of the 11th Text Retrieval Conference, 591-594.