# Fujitsu Laboratories TREC2001 Report

Isao Namba

Fujitsu Laboratories Ltd.
namba@jp.fujitsu.com

**Abstract**

This year a Fujitsu Laboratory team participated in web tracks. Both for ad hoc task, and entry point search task, we combined the score of normal ranking search and that of page ranking techniques. For ad hoc style task, the effect of page ranking was very limitted. We only got very little improvement for title field search, and the page rank was not effective for description, and narrative field search.

For entry point search task, we compared three heuristics. The first heuristics supposed that entry point page contains key word and had high page rank score. The second heuristics supposed that entry point page contains key word in its head part and had high page ran score. The third heuristics supposes that entry point is pointed by the pages whose anchor string contains key word, and has high page rank score. The page rank improved the result of entry point search about 20-30% in rather small VLC10 test set, and the first heuristics got the best result because of its high recall.

## 1 System Description

For TREC2001, we added the new functions to trec_exec for entry point search. The functions includes score merging, evaluation of reciprocal rank and so on. We used Web Recommener Agent to get page ranking score. Except above modifications, the framework is same as that of TREC9[1].

#### 1.0.1 Teraß

Teraß[2] is a fulltext search library, designed to provide an adequate number of efficient functions for commercial service, and to provide parameter combination testing and easy extension for experiments in IR.

#### 1.0.2 trec_exec

trec_exec is designed for automatic processing of TREC. It contains a procedure controller, evaluation module , logging module, and all non-searching units such as query generation, query expansion and so on. trec_exec can execute all the TREC processing for one run in a few minutes, and it can be used for system tuning by hill-climing. The new functions added for TREC2001 Web track are heuristics for entry point search, evaluation of reciprocal rank, and accepting non-digit query number.

#### 1.0.3 Web Recommender Agent

We used web recommeder agent tool developed for automatic domain specific web directory Tsuda et al[3] to get page ranking score. The page rank score is put into Teraß, and it is marged with normal ranking score.

# 2 Common Processing

## 2.1 Indexing/Query Processing

### 2.1.1 indexing vocabulary

The indexing vocabulary consists of character strings made up of letters, numbers, and symbols, and no stop words were used in indexing. For TREC8, we modified the grammar of the token recognizer to accept acronyms with symbols such as U.S., and AT&T as one token.

### 2.1.2 Stemmer

As the experiment in TREC8[1] shows, SMART[4] stemmer seems to be stable, we used SMART.

### 2.1.3 Information in inverted file

Text number, term frequency, and term position are stored for run time phrase processing.

### 2.1.4 Stop word list for query processing

As in the TREC8[1], we used a stop word list of about 400 words of Fox[5], and words with a high df (more than 1/7 of the number of all documents) were also treated as stop words.

### 2.1.5 Stop pattern removal

The expression of TREC queries are artificial, so frequently appearing patterns such as "relevant document" are stop patterns. We generalized this observation, and removed the words which meet one of the following condition.

1. Word in stopword list is a stopword.
2. Word which is not a proper noun[1], and whose df in TREC1-7 queries is more than 400*0.1 is a stop word.
3. Word bi-gram whose df in TREC1-7 queries is more than 400*0.02 is a stop pattern.
4. Word tri-gram whose df in TREC1-7 queries is more than 400*0.01 is a stop pattern.
5. All the words in a sentence that contains "not relevant" are stop words.
6. 4 words following "other than" are stop words.
7. 4 words following "apart from" are stop words.

## 2.2 Weighting Scheme for Ranking

The scheme for term weight is

$$\begin{aligned} RankScore_of_term(term) &= qtf * tf * idf \\ RankScore(t) &= \sum RankScore_of_term() \end{aligned} \tag{1}$$

where

*qtf* is query term weight, *tf* is term weight in document, *idf* is inverse document frequency, and *t* is document. The score for one document is the sum of the term weights with co-occurence boosting.

[1] U.S appears 94 times in TREC1-7 queries.

1. qtf

   qtf is the combination of the following parameters

   $$qtf = \sum_f fw * tf * ttw \tag{2}$$

   where

   $f$ is the topic field (title, description or narrative).
   $fw$ is weight of the topic field.

   We set the value for the title field to 1.0, the value for the description field 1.0, the value for the narrative is 0.7. The weight for title field is decreased for TREC2001 because weighting title field that is weighting raw Web query, does not produce good result.

   Some teams [6], [7],[8] used weighting depending on field type, and we take the same approach.

   $tf$ is the bare frequency in each field.

   $ttw$ is the term type weight. It is set to 3 for terms, and set to 1 for phrase(word bi-gram).

2. tf

   We simply used the tf part of OKAPI[6].

   $$tf = \frac{(k_1 + 1) * term\_freq}{(k_1((1-b) + \frac{b*doc\_length\_in\_byte}{average\_doc\_length\_in\_byte})} \tag{3}$$

   where $k_1 \; = 1.5, b = 0.75$

3. idf

   We used a modified idf of OKAPI. We introduced a cut off point for low df words, and decreased the idf value for high df words.

   $$idf = log_2 \frac{N - (n * \alpha)}{n} \tag{4}$$

   where

   $N$ is the number of documents
   $n$ is document frequency(df) if df > 1/10000 * N else $n$ is $1/10000 * N$
   $\alpha$ is set to 3

## 2.3 Co-occurence Boosting

As in TREC8, we use co-occurence boosting techinique which favours co-occurence of query terms in a document. Co-ocurrence boosting is implemented by simply multipling the boost ratio to the similarity of each term.

$$S_i = \sum_t B * W_{t,i} \tag{5}$$

where
$S_i$ is the degree of similarity between a document and topics.
$i$ is the document number.
$t$ is a term that document$_i$ includes.
$W_{t,i}$ is the part of similarity of term$_t$ in document$_i$.
$B$ is the boost-ratio by term co-occurrence.

The best parameter $B$ depends on the query, but it is difficult to tune them for each query. As in the case of field weighting, the weight for title field is decreased. We set the $B$ to 1.05 for the title word, to 1.05 for the description word, and to 1.00 for the narrative word, and to 1.0 for the word added by query expansion.

## 2.4 phrase(bi-gram)

Instead of traditional IR phrase (two adjacent non-stopword pair with order or without order), we permitted limited distance in phrase. The motivation for introducing fixed distance is that that non-stopword may exist between two adjacent words in a query, and it producued slightly better result in the past experiment.[1] The term weight of bi-gram is fixed as 1/3 of a single word, and the distance is set to 4.

The phrase(bi-gram) is not used for entry point search, as it was too restrictive.

## 2.5 Query Expansion

Query Expansion was used for the ad hoc task, and small web track. The Boughanem formula[6] was used to select terms.

$$TSV = (r/R - \alpha s/S).w^{(1)} \tag{6}$$

where

$w^{(1)}$ is modified and more general version of Robertson/Sparck Jones weight.

The $\alpha$ was set 0.001, and k4 was -0.3, k5 was 1, and k6 was 64. The top 20 documents in the pilot search were supposed to be relevant, and the documents ranked from 500 to 1000 were supposed to be non-relevant. The top ranked 40 words which are not included in original query, which are not included in the stopword list of SMART, whose tsv score are more than 0.003, whose df are more than 60, and whose df are less than 200000 were added to the original query.

No collection enrichment technique was used, and query expansion was used only for ad hoc runs.

## 2.6 Page Ranking

Google is famous search engine that uses link based ranking approach[9]. The intutive idea of Google is that pages cited frequently are important, and that pages cited from important pages are also important. We adopted a Revised Page Ranking scheme which is proposed in Tsuda et al[3]. The scheme distingues the internal server(local) link, and external server(remote) link. The modification reflects the fact that the local link may be self link and less important than the link from external server (linked from others).

$$\begin{aligned} PageRank(A) &= (1-d) + d * \frac{PageRank(T_i)}{RC(T_i, A)} \\ RC(T, A) &= \frac{C(T)^2}{C_{rem}(T) + \alpha C_{loc}(T)} \\ &\quad (T, A : different\_domains) \\ \\ &= \frac{\alpha C(T)^2}{C_{rem}(T) + \alpha C_{loc}(T)} \\ &\quad (T, A : same\_domains) \end{aligned} \tag{7}$$

where

$C_{rem}(T)$ is the number of remote link from $T$

$C_{loc}(T)$ is the number of local link from $T$

$C(T) = C_{rem}(T) + C_{loc}(T)$

$\alpha[0, 1]$ is weighting factor for local link.

The $d$ is set to 0.5, and the local link factor $\alpha$ is set to 0.1 for official runs.

## 2.7 Marging Score and Reranking

Both for entry point search, and ad hoc search of title field query, top $N$ doucments are retrieved by normal ranking strategy first, and the documents are resorted by using page ranking score. To merge the normal ranking score and page ranking score, we levelize their gap by comparing their average score in top $N$ documents. The equation8 is used for reranking.

$$\begin{aligned} S(t) &= (1-\alpha) * gap * RankScore(t) \\ &\quad + \alpha * PageRank(t) \\ gap &= \frac{\sum_{i=1}^{n} PageRank(i)}{\sum_{i=1}^{n} RankScore(i)} \end{aligned} \tag{8}$$

where

$RankScore$ is score of ranking for a document
$PageRank$ is score of Page rank for a document
$\alpha$ is RankScore factor which takes between 0 and 1
$n$ is the number of TREC output or the number of document retrieved.

The $RankScore$ factor is different for ad hoc search, and entry point search. For ad hoc search, $RankScore$ facotor is set to 0.95 or 1.0. It is because we got no improvment for ad hoc search except in title field using $PageRank$ score. For entry point search, $RankScore$ facotor is set to 0.47.

# 3 Ad hoc Search

Except title only runs, the query processing is same as that of traditional ad hoc task.

## 3.1 Result

Four runs were submitted, ie. flabxt, flabxtl, flabxtd, and flabxtdn. In the run id, the infix 'l' means link, 't' means using title field, 'd' means using description field, and 'n' means using narrative field.

| Name | flabxt | flabxtl | flabxtd | flabxtdn |
|---|---|---|---|---|
| field | T | T | TD | TDN |
| link | NO | YES | NO | NO |
| Average Prec | .171 | .170 | .233 | .184 |
| R-Prec | .218 | .208 | .261 | .224 |
| P@20 | .279 | .277 | .355 | .316 |
| Retrieved | 50000 | 50000 | 50000 | 50000 |
| Rel-ret | 2155 | 2151 | 2449 | 2170 |
| Relevant | 3363 | 3363 | 3363 | 3363 |

Table 1: Official ad hoc result

The effect of page ranking is very limited or obscure for ad hoc search. We get very little improvement only for title only field search for test run, and no improvement for description, narrative field search at all.

It seems that web page with high page ranking score is often top of domain, or user, and is informative, but does not necessarly match the information need of ad hoc style query.

# 4 Entry Point Search

For all entry point search runs, we used characteristics of Web, that is page rank score, anchor string and document structure.

## 4.1 Heuristics for entry point search

For entry search we experimented three different heuristics. We describe them here.

1. Simple Page Rank

   The first heuristics supposes that good entry point contains key words (theme of page) in it and has high page rank.

   This approach seems to be popular in web search engines such as google, teoma[10], and wisenut[11], and to produce good result if compared with simple ranking search.

   The ranking proecedure is that top 1000 pages are ranked by ranking equation1 , and they are rerankked using equation8.

2. Head Part and Page Rank

   The title of Web page often appears to contain key words (theme of page). For example, the entry point page for EP5 query "Haas Business School" contains "Haas School of Business"in head part, and the entry page for EP2 "Hunt Memorial Library" contains "Hunt Libary" in head part.

   The second heuristics supposes that good entry point contains key words (them of page) in head part of the page, and has high page rank score.

   As the head part of the page, we used top 256 byte of each page.

   This heuritstics might not get better result than simple page ranking, but was expected to get high precision if head part contains keywords.

   The ranking procedure is the same as that of simple page ranking heuristics.

3. Pointed by Anchor and Page Rank

   Web page name which is in anchor string seems to be most direct and reliable evidence of entry page though anchor string often contains pronoun such as "this", "here". Third heuristics suppose that good entry point is pointed by anchor string of high ranked pages, and has high page rank.

   In our experiment, we use 75 byte string around anchor, instead of using just anchor string. It is because we got little avaliable anchor string set. If WT10G test set contains enough web pages whose out link contains anchor string to the pages within WT10G, and that anchor strings match entry point search query, this heuristics is expected to be best in the three.

   The searching procedure is as follows.

   (a) Searching anchor string (around anchor string 75byte), and reraking using equation8. (anchoring page)

   (b) Collecting document ids which is pointed out by anchoring page.(referred page)

   (c) Scoring the referred page by following equation.

$$\begin{aligned} Ref(t) &= (1-\alpha) * gap * PageRank(t) + \\ &\qquad \alpha * ReferScore(t) \\ ReferScore(t) &= \sum_{i=0}^{n} Score(i) \\ Score(i) &= Rank(max)/Rank(i) * S(i) \end{aligned} \tag{9}$$

   where

   $t$ is a document id in referred page set.

   $\alpha$ is Page Rank factor which takes between 0 and 1.

$i$ is a document id of anchoring page which referts document id $t$
$Rank$ is rank of document id.
$max$ is max number of retrieved text of anchoring page search.
$S(i)$ is equation8.

### 4.2 Result

Four runs were submitted:flabxeall, flabxet256, flabxe75a, and flabxemerge. flabxeall used Simple Page Rank1 heuristics, flabxet256 used Head Part and Page Rank2 heuristics, flabxe75a used Pointed by Anchor and Page Rank3, and flaxemerge merged the result of flabxeall, flabxet256, and flabxe75a. The table2 also includes entry point search without page ranking for comparison.

| Name | flabxeall | flabxet256 | flabxe75a | flaxemerge |
|---|---|---|---|---|
| Relevant | 145 | 145 | 145 | 145 |
| Retrieved@100 | 131 | 96 | 90 | 96 |
| Retrieved@10 | 117 | 73 | 81 | 74 |
| Rec-rank@100 | .599 | .363 | .399 | .363 |

Table 2: Entry Point Search Result

## 5 Conclusion

For ad hoc style search, we did not get improvment by just combing normal ranking score and page ranking score. But it is uncertain whether page ranking score has no effect for ad hoc style search, or WT10G test set is too small for ad hoc search using page ranking. For entry page search, we get about 30% improvement using page ranking score.

## Acknowledgment

We thank to Dr. Tsuda who prepared page ranking score for WT10G test set.

## References

[1] I Namba and N Igata. Fujitsu laboratories trec8 report. *The Eighth Text REtrieval Conference*, 2000.

[2] I Namba, N Igata, H Horai, K Nitta, and K Matsui. Fujitsu laboratories trec7 report. *The Seventh Text REtrieval Conference*, 1999.

[3] Hiroshi Tsuda, Takanori Ugai, and Kazuo Misue. Link-based acuqistion of web metadata for domain-specific directories. *The 2000 Pacific Rim Knowledge Acquistion Workshop(PKAW2000)*, 2000.

[4] SMART ftp cite. ftp://ftp.cs.cornell.edu/pub/smart/. 1999.

[5] Chiristopher Fox. Chapter 7, lexical analysis and stoplists. *Information Retrieval Data Structure and Algorithms ed. William B. Frakes, Ricardo Baeza-Yates Prentice Hall*, 1992.

[6] S E Robertson, S Walker, and M Beaulieu. Okapi at trec-7. *The Seventh Text REtrieval Conference*, 1999.

[7] D R H Miller, T Leek, and R M Schwarts. Bbn at trec-7. *The Seventh Text REtrieval Conference*, 1999.

[8] James Allan, Jamie Callan, Mark Sanderson, Jinxi Xu, and Steven Wegmann. Inquery and trec-7. *The Seventh Text REtrieval Conference*, 1999.

[9] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *WWW7 Conference*, 1998.

[10] teoma. http://www.teoma.com. 2000.

[11] wisenut. http://www.wisenut.com. 2000.