IIT at TREC-2003

Task Classification & Document Structure for Known-Item Search

Steve Beitzel, Eric Jensen,
Rebecca Cathey, Ling Ma
David Grossman, Ophir Frieder
{steve, ej, cathey, maling, grossman, ophir}@ir.iit.edu

Information Retrieval Lab
Department of Computer Science
Illinois Institute of Technology
Chicago, Illinois

Abdur Chowdhury, Greg Pass,
Herman Vandermolen
{cabdur, gregpass1}@aol.com,
herman@aol.net

Search Technologies Group
America Online, Inc.
Dulles, VA

Abstract:
This year's TREC 2003 web task incorporated two retrieval tasks into a single set of experiments for Known-Item retrieval. We hypothesized that not all retrieval tasks should use the same retrieval approach when a single search entry point is used. We applied task classifiers on top of traditional web retrieval approaches. Our traditional retrieval is based on fusion of result sets generated by query runs over independent parts of the document structure. Our task classifiers combine query term analysis with known information resources and URL depth. This approach to task classification shows promise: our classified runs improved overall MRR effectiveness over our traditional retrieval results by ~10%; provided an MRR of .665; ranked 87% of relevant results in the top 10; correctly ranked the #1result 56% of the time. 67% of the queries performed above the average, and 49% above the median.

Keywords: Known-item search, document structure retrieval, query task classification

Introduction

Many years of research have been devoted to examining the question of what is the best retrieval strategy for retrieving information. This year we explore a variation on the task in which a specific home/named page or known-item is sought after given a query or topic. Our research this year builds on prior known-item and homepage retrieval techniques by examining the question of whether these two tasks should be treated differently.

Basic retrieval work has focused on ranking strategies: for example, some of the most studied algorithms include PDLN (Pivoted Document Length Normalization) [1], Okapi BM25 [2], Self-Relevance [3], and Language Models [18]. All these ranking strategies try and find better ways to estimate relevance, as do many of the newer language models. In our tests, BM25 has consistently outperformed the other strategies, so we use it in our experiments.

Web retrieval extends basic full-text retrieval by using link and document structures to provide various document representations [4]. This multi-document representation approach was shown to be effective in the top web track systems at the 2002 TREC conference. The basic hypothesis is that content developers use HTML elements/tags to improve the readability of their documents, thus using that information during the ranking process via multiple document representations will improve effectiveness. Examples of these representations could be title, section headers, anchor text, bold, underlines, comments, referring page anchor text, etc. We initially focus on title, anchor text, and referring anchor text.

Given multiple document representations, the most fitting method of using and combining those representations for a given query becomes a research question. In recent years, the category of work known as data fusion, or multiple-evidence, describes a range of techniques in information retrieval whereby multiple pieces of information are combined to achieve improvements in retrieval effectiveness. These pieces of information can take many forms including different query representations, different document representations, and different retrieval strategies used to obtain a measure of relationship between a query and a document.

Several researchers have used combinations of different retrieval strategies to varying degrees of success in their systems [5, 6]. Belkin et al. examined the effects of combining several different query representations to achieve improvements in effectiveness [7, 8]. Lee examined the effect of using different weighting schemes to retrieve different sets of documents using a single query and document representation, and a single retrieval strategy [9]. Fox and Shaw examined combination algorithms that increase the score of a document based on repeated evidence of its relevance in [5].

One of the algorithms designed by Fox and Shaw, CombMNZ, has proven to be a simple, effective method for combining result sets. It was used by Lee in his fusion experiments, and has become the standard by which newly developed result combination algorithms are judged. More recent research in the area of meta-search engines has led to the proposal of several new result combination algorithms [10, 11, 12]. Although these algorithms were shown to be comparable, and on occasion superior, to CombMNZ, we use the widely-used CombMNZ for this work, leaving other approaches as a topic of further research.

Our traditional web search approach fuses the results from different document structure indices to produce a single ranked list for the known-item task. The results were fused using linear combinations based on estimated MRR values in order to maximize mutual evidence [13].

In the next section we describe our basic search approach in more detail. In the task classification section we present our approach to using task information to improving task and overall system effectiveness. Lastly, we present our experimental results and conclude with future possible research directions.

1 Traditional Search Approach

To conduct our research we use the IIT retrieval system AIRE (http://ir.iit.edu/projects/AIRE.html) [14]. This system builds a traditional inverted index based on a given document structure(s). For stemming, our system uses conflation classes [15] instead of a more commonly used stemmer such as Porter [16]. Those classes have been modified over the years as problem term variants have been encountered. Additionally, AIRE uses a generated statistical phrase list, where the statistical phrases were generated with a news collection and IDF filtering to reduce the final phrase list size. Phrases are generated via a bi-gram sliding window algorithm and weighted with 25% importance in relation to keyword weighting for retrieval. Basic term weighting uses the Okapi BM25, Equation 1.

$$\sum \log\left(\frac{(N-n)+.5}{(n+.5)}\right) * \left(\frac{(k1+1)*tf}{(K+tf)} * \frac{(k3+1)*qtf}{(k3+qtf)}\right)$$

$$K = k1*((1-b)+b*dl/avdl)$$

Equation 1: Okapi BM25

Where:

- tf = frequency of occurrences of the term in the document
- qtf = frequency of occurrences of the term in the query

- dl = document length
- avdl = average document length
- N = is the number of documents in the collection
- n = is the number of documents containing the word
- k1 = 1.2
- b = 0.75 or 0.25 (we use .75 for full text and .25 for shorter representations, see appendix)
- k3 = 7, set to 7 or 1000, controls the effect of the query term frequency on the weight.

1.1 Parsing

We indexed the 18GB .GOV collection producing a full-text index, an HTML title term index, and an anchor text index. The anchor text index differed from the other indexes, in that an additional mapping stage was required so referencing anchor text data can be linked to the referenced TREC document name. For our experimental layout we first produced a baseline run using BM25, conflation classes, phrases, and full-text indexing (referred to as the "Full text" run in the results summarized in Table 1).

1.2 Fusion

Our linear combination consists of the following steps. First, for each document representation retrieved, the scores are normalized using exponential z-score normalization, as in Equation 3. The advantage of this method is that it preserves all relationships of the values exactly; it does not introduce any potential bias into the data. The final scores are calculated using CombMNZ, as in Equation 2, where each individual score is biased via weights assigned to the document structure by prior MRR estimates.

$$\boxed{\text{CombMNZ} = \text{SUM(Individual Similarities)} * \text{Number of Nonzero Similarities}}$$

Equation 2: CombMNZ

$$\boxed{\mathit{norm\_score}(d,x) = ((e^{\wedge}(\mathit{orig\_score}(d)) - \mathit{mean}(x)) / \mathit{stddev}(x))}$$

Equation 3: Exponential Z-Score Normalization for document d and document representation x

2 Task Classification

To explore our hypothesis, we identify home pages via two techniques. The first technique uses known information resources and seeks to match those resources to queries. The second approach classifies queries based on keywords like “homepage”, and then uses probability distributions of URL length to improve the classification.

2.1 Known-Resource Matching

As many of the homepages in the .GOV domain are government agencies, we hypothesized that simply pairing queries with homepages by matching names and acronyms of agencies would be effective. We searched the web for lists of government agencies and their associated acronyms and homepages, choosing http://www.ulib.iupui.edu/subjectareas/gov/docs_abbrev.html because it provided all three pieces of information, was reasonably large, and was easy to parse with a simple regular expression.

We matched queries to this parsed list of agency name, acronym, and URL tuples using the matching algorithm below. Our system matched 26 of the 300 queries, found the correct homepage for 24 of them, improving our results for 11 queries over our traditional web approach combined with URL normalization. We combined these matching homepages with the final result sets by simply inserting them at rank one. Of the matching queries, 13 already had the matched result at rank one in our final fused, URL length-weighted result set and 2 had not previously been found in the top 1000 results. The other 11 queries matched the relevant homepage, so inserting that homepage at the first result instead of its previous lower position in the result set offered an improvement. In total, MRR was improved by 0.05. Our complete known-resource matching algorithm is shown in Figure 1.

Known-Resource Matching Algorithm:

Step 1. Strip "home", "homepage", and "page" from the query. Strip "the" if it appears as the first word.

Step 2. If the remaining query is an acronym (any sequence of capital letters and spaces), look it up in the list of acronyms by case-insensitive exact string matching. Else, remove any acronyms that might be present alongside other terms from the query, normalize the spacing in the query, and look it up in the list of agency names by case-insensitive exact string matching.

Step 3. If we found a matching acronym or agency name, convert its URL to a canonical form by stripping "http://", "www", trailing slashes, etc. and look it up in a list of all the URLs in the .GOV by case-insensitive exact string matching.

Step 4. If we found a matching acronym, but could not find its corresponding URL in the .GOV, look for its corresponding URL with the last path element stripped off and just the matching acronym as "http://www.ACRONYM.gov" in the .GOV

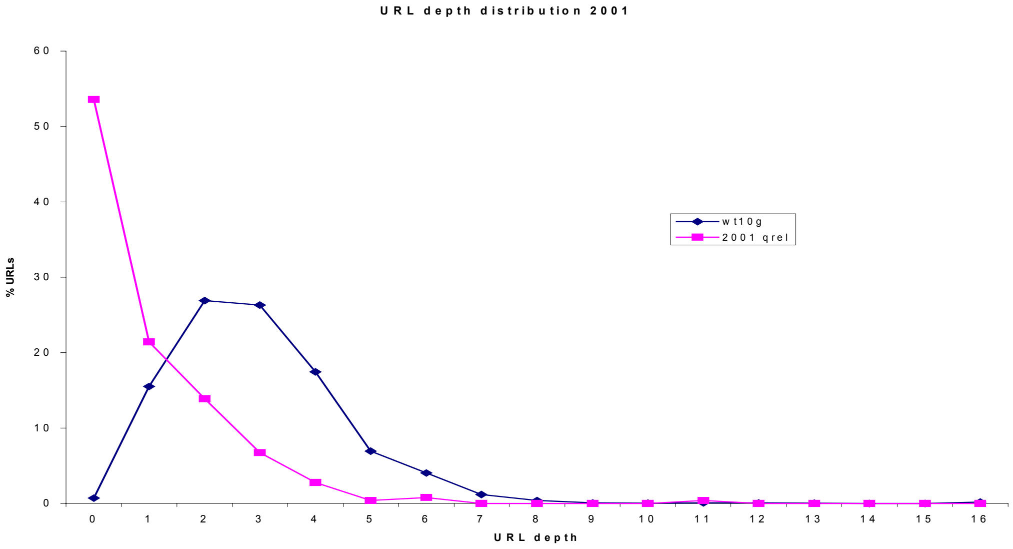
Figure 1: Known-Resource Matching Algorithm

2.2 *Task Classification*

Kraaij, Westerveld, and Hiemstra [17] previously examined differences in the distributions of URL depth (the length of the path in the URL) between known home pages (from TREC-2001 answers) and the WT10g test collection. They showed that these distributions were very different, and that this could be used to improve the ranking of the results for home page queries. Thus it appeared that if we would be able to successfully classify queries as either home page queries or something else (named page queries in this case), we should be able to improve the results for the homepage queries.

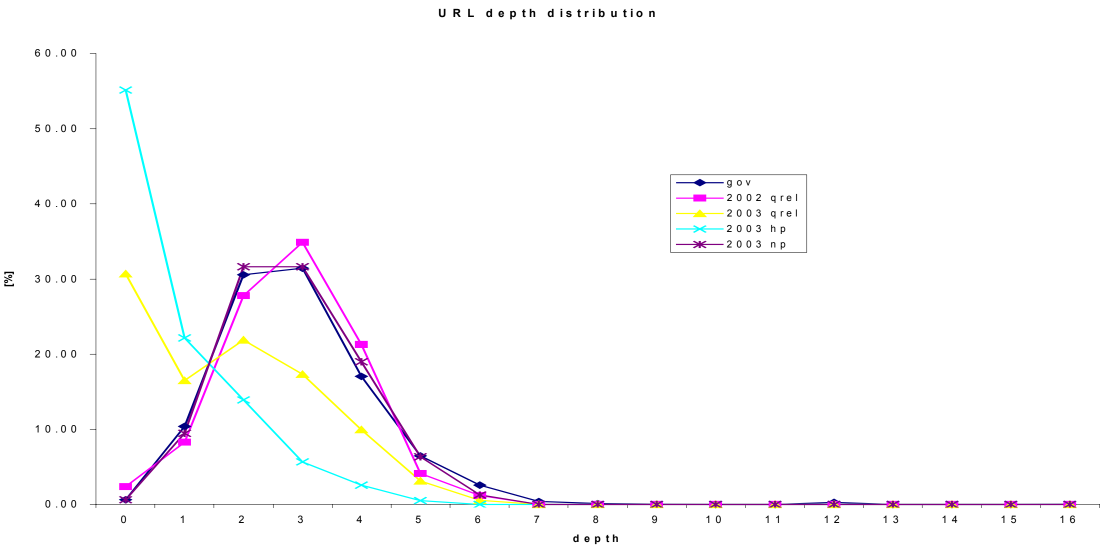
We used a definition of URL depth that was slightly different from the one used by Kraaij et al. but confirmed the differences in distributions. We removed from the URL the leading parts, including host, domain, port, etc., up to the path. We then removed trailing occurrences of "index.htm" and "index.html", and counted the number of path elements remaining to determine the URL depth. The graph below shows the URL depth distribution for the WT10g collection and the correct answers for the TREC-2001 homepage task.

Table 1: WT10g collection URL distributions



For TREC-2003 we ran the same analysis against the .GOV collection and the now known correct answers (qrels) for both the homepage queries and the named-page queries. The analysis shows that for homepage queries, the same distribution differences can be seen between the correct answers and the collection as a whole that were observed in TREC-2001. In addition, it shows that the URL depth distributions for the named page query results are virtually identical to that of the collection as a whole, and thus no advantage can be gained for named page queries.

Table 2: GOV collection URL Distribution



To determine if there were other variables we could utilize, we examined last-modified-date, and in-domain and out-domain link information, and found no significant difference in distributions for the correct answers versus the .GOV collection as a whole.

To be able to take advantage of the URL depth information for home page queries without disturbing the rankings for the named page queries, we attempted to classify the queries into one of these two groups. We created a list of 32 keywords that we believed were good indicators of a home page query. This list includes words like “home”, “homepage”, “administration”, “agency”, etc. Some of these terms were generic, but many would likely be specific to the GOV collection. We parsed the queries looking for these words. Our algorithm categorized 108 (36%) queries (out of 300 combined) as home page queries. Of those 108, 15 were false positives, and 93 were correctly classified. Of the 150 home page queries in the query set, 57 did not match any of the criteria in the classifier and were not marked as home page queries (false negatives).

We took the results from the fusion run and modified the scores of the documents for those queries that our classifier marked as home page queries. The algorithm for the score boosting is shown in Equation 4:

$$\boxed{S_i^* = S_i + \alpha P(d_i)}$$

Equation 4: Score Boosting Formula

where S_i^* is the newly assigned score of document i, S_i is the original score of document i (after fusion), α is a constant, d_i is the URL depth of document i, and $P(d_i)$ is the probability that a document would have

URL depth d_i if it was given that it was a home page. After some experimentation we set the value of α to 12.

3 Results

Our approach is to build on prior approaches to known-item retrieval. To that end, we first examined the effectiveness of a full text approach based solely on BM25 ranking. In the following table we see that our estimated (.52) and actual (.42) effectiveness of this approach can be improved.

We next followed the structured approach that others have shown to be effective by exploiting HTML structure. In the second set of experiments we fused the title, anchor, and full text indices with the CombMNZ algorithm with linear weighting based on estimated MRR values. The Appendix displays the results of those experiments; while we did not have the final qrels, our estimated qrels provided equivalent results to real probabilities. The overall improvement of using document structure over full text retrieval by using CombMNZ with MRR linear combinations improved our effectiveness by 42%.

We next examined our use of known resource information to our traditional web based search approach. By using our known resource information that is based on that task, our MRR is .65.

Next, we examined our classification approach over prior web techniques. With the usage of prior probability factoring with our task classifier, we improved the effectiveness of the system by 3% estimated and 4% actually. We then examined this effectiveness assuming a perfect classifier, and found that our MRR increased to .663, or an additional 4% improvement.

Finally, we examined the improvements of combining both our known resource and URL factoring on the overall effectiveness: we found that by combining those approaches our MMR was raised to .665 and with a perfect classifier .685, an improvement of 9% over our fused results and 55% over our full text results.

Features	Run Tag	Description	Training MRR	Actual MRR	W/ Perfect Classifier & P Dist
Full text	iit03wp75	Full text using statistical phrases weighted at 0.25, BM25 with b=0.75	.52	.43	n/a
Fusion	Iit03wtaez	CombMNZ(fulltext b=0.75, title b=0.25, anchor b=0.25) Using Z-Score and Exponential Normalization	.62	.61	n/a
Fusion, *Known-Resources*	iit03sa	Same fusion, insert known resources with matching names or acronyms at first position	.6889	.65	.65
Fusion, *URL Length Weighting*	iit03su	Same fusion, re-weight results using prior probabilities of relevance given URL lengths calculated by maximum likelihood of training qrels	.6430	.636	.663
Fusion, *URL Length Weighting,* *Known-Resources*	iit03sau	Same fusion, same re-weighting based on URL length priors, and same known-resource insertion	.6945	.665	.685

Table 1: Submitted Runs

	iit03wp75	iit03wtaez	iit03sa	iit03su	iit03sau
MRR	.443	.611	.651	.636	.665
In Top 10	.67	.84	.867	.857	.87
Not Found	.197	.087	.073	.08	.07
=>Median	.253	.44	.47	.47	.49
>= Mean	.407	.613	.653	.647	.67

Table 2: Submitted Runs Official Evaluation

Our final iit03sau approach for the known-item task was 49% of the time equal or above the median and 67% above the mean score of submitted runs. Additionally, our approach produced the item in the top 10 results 87% of the time and only missed 7% of the results.

These results provide validation of the robustness of our task algorithm; more research needs to be conducted to find other task specific information to determine how that information should be incorporated into the ranking strategy.

4 Preliminary Failure Analysis

Examining Table 2 gives some indication of how each approach performs. Notice for the iit03sa and iit03sau runs that relevant documents are not found for approximately 7% of the queries, and are found in the top ten for 87% of the queries, leaving 6% for which the relevant document is found, but is poorly ranked. For failure analysis we focus on these runs, as they achieve the highest MRR.

We examined the queries where the relevant document was not found or poorly ranked (found, but not in the top ten). For each of these queries from the iit03sa and iit03sau runs, we examined the relevant documents and noted where in the each document the query text was present. Our hope was that this analysis would give us an idea of which of our document representations was most likely to contain query text for queries that performed poorly, and we could use that knowledge to improve our parsers. This analysis is shown in Table 3, where the left side represents queries where the relevant document was poorly ranked, and the right side represents queries for which the relevant document was missing.

From Table 3 we see that for queries where the relevant document is missing or poorly ranked, the query text appears within the relevant document approximately 95% of the time. Since the title is most likely to contain the query text (72-82%), it is reasonable to conclude that our title parser might be failing often on these documents. Query text is also likely to appear in the body of the document (56-62%), and particularly in the anchor text for queries where the relevant document is missing (50-52%). From these results we conclude that our title and body parsers may be at fault and worthy of further examination.

To examine this further, we attempted to determine where our parsers were failing, paying particular attention to the anchor text and title parsers. We noticed that although the title parser extracts most of the titles, there are several instances where it fails for various idiosyncratic reasons. Likewise, the anchor text parser also worked fairly well, extracting all the data from within the <a> tags. However, it did not extract any data where the "href" parameter of the <a> tag was pointing to itself. For example, when the "#" sign is used with hyperlinks it is usually followed by redirection within the page such as "top" or "back". However, in some cases, there are hyperlinks with "#" sign followed by meaningful text. In the case of "<a href = "#3214">u.s.s. monitor </a>", our anchor text parser would ignore "u.s.s. monitor". By improving the efficiency of both the title and anchor text parsers, we believe the accuracy of each individual result file can be improved, thus improving the accuracy of the final results file.

We also did some analysis to try and determine whether our fusion process was causing relevant documents to be pushed down in the result set to poor ranks. To test this, we examined the three result files used in the fusion process: iit03wp75, iit03t_np, and iit03a_np. Once again we examined two cases: poorly ranked documents and missing documents. In the final result file there are 18 queries that perform poorly and 21 queries that don't return relevant documents. For each of these queries, we calculated the distribution of relevant documents over each individual result file. In addition, we computed the percentage of relevant documents whose ranks occurred in the top ten, in between ten and fifty, and over fifty. The results of this analysis are given in Table 4.

For queries with poor performance, relevant documents are most likely to occur in the title (45%) and word (66%) result files. Overall, 50% of the relevant documents occurred without overlap in the top ten of the individual result files, and the fusion process increases these ranks, thereby damaging overall MRR.

Similarly, for queries whose relevant documents are missing, the word and title result files contain a high percentage of relevant documents. Unfortunately in this case the majority of the relevant documents occurring in the word and title result files have ranks over fifty prior to fusion, therefore, performance on these queries was initially bad and the fusion process is not likely to have had a significantly negative impact in this case. From the analysis we can conclude that a more finely tuned fusion method may have the potential to help cases where the relevant documents end up poorly ranked in the final result set.

	Poorly Ranked		**Missing**	
	iit03sa	**Iit03sau**	**Iit03sa**	**iit03sau**
Title	.72	.72	.82	.81
Word	.56	.56	.59	.62
ImgAlt	.44	.39	.45	.48
Anchortext	.39	.39	.50	.52
Meta	.28	.28	.14	.10
Not Found	.06	.06	.05	.05

Table 3: Presence of Query Terms in Documents

Rank in resultfile	**Poorly Ranked**			**Missing**		
	iit03wp75	**iit03t_np**	**Iit03a_np**	**iit03wp75**	**iit03t_np**	**iit03a_np**
top 10	.22	.17	.11	.05	.0	.0
10-50	.44	.28	.22	.10	.10	.0
over 50	.28	.28	.0	.57	.48	.29
total	.94	.73	.33	.72	.58	.29

Table 4: Presence of Relevant Documents in Individual Result files

5 Conclusion

This year we participated in the homepage and known-item web retrieval task. We explored the concept of multiple tasks being issued via the same interface. To that end we explored using a task classification approach where we could use task specific information to improve those queries. This approached showed promise in that by using task specific information our results improved ~10% over our baseline traditional web retrieval approach and would have improved by 12% given an optimal classifier. Given the simplicity of our classifier this approach seems to help the overall system effectiveness. Our future work will continue examining other features that can help in the other tasks.

Appendix

Table 5: B-value = .25

Index	Run Description	Run Name	Est. MRR	Actual MRR
gov.anchor	.GOV anchor terms only, good HTML parser, no phrases	iit03a_np.dat	0.24	0.29
gov.anchor	.GOV anchor terms only, good HTML parser, with phrases	iit03a_p.dat	0.24	0.30
gov.title	.GOV title terms only, good HTML parser, no phrases	iit03t_np.dat	0.34	0.33
gov.title	.GOV title terms only, good HTML parser, with phrases	iit03t_p.dat	0.35	0.34
gov.bigtext	.GOV bigtext terms only, good HTML parser, no phrases	iit03b_np.dat	0.18	0.15
gov.bigtext	.GOV bigtext terms only, good HTML parser, with phrases	iit03b_p.dat	0.18	0.15
gov.word	.GOV conglomerate, Words only, good HTML parser, no phrases	iit03w_np.dat	0.34	0.29

gov.word	.GOV conglomerate, Words only, good HTML parser, with phrases	iit03w_p.dat	0.34	0.29
gov.meta	.GOV meta, Words only, good HTML parser, no phrases	iit03m_np.dat	0.17	0.14
gov.meta	.GOV meta, Words only, good HTML parser, with phrases	iit03m_p.dat	0.17	0.14
gov.imgalt	.GOV img/alt, Words only, good HTML parser, no phrases	iit03i_np.dat	0.16	0.16
gov.imgalt	.GOV img/alt, Words only, good HTML parser, with phrases	iit03i_p.dat	0.15	0.16

Table 6: B-value = .75

Index	Run Description	Run Name	est. MRR	actual MRR
gov.anchor	.GOV anchor terms only, good HTML parser, no phrases	iit03a_np75.dat	0.29	0.33
gov.anchor	.GOV anchor terms only, good HTML parser, with phrases	iit03a_p75.dat	0.29	0.33
gov.title	.GOV title terms only, good HTML parser, no phrases	iit03t_np75.dat	0.30	0.26
gov.title	.GOV title terms only, good HTML parser, with phrases	iit03t_p75.dat	0.30	0.26
gov.bigtext	.GOV bigtext terms only, good HTML parser, no phrases	iit03b_np75.dat	0.18	0.16
gov.bigtext	.GOV bigtext terms only, good HTML parser, with phrases	iit03b_p75.dat	0.18	0.16
gov.word	.GOV conglomerate, Words only, good HTML parser, no phrases	iit03w_np75.dat	0.49	0.42
gov.word	.GOV conglomerate, Words only, good HTML parser, with phrases	iit03w_p75.dat	0.52	0.43
gov.meta	.GOV meta, Words only, good HTML parser, no phrases	iit03m_np75.dat	0.12	0.09
gov.meta	.GOV meta, Words only, good HTML parser, with phrases	iit03m_p75.dat	0.11	0.09
gov.imgalt	.GOV img/alt, Words only, good HTML parser, no phrases	iit03i_np75.dat	0.12	0.11
gov.imgalt	.GOV img/alt, Words only, good HTML parser, with phrases	iit03i_p75.dat	0.12	0.11

References

[1] A. Singhal, et al., "Pivoted document length normalization", ACM-SIGIR, 1996.

[2] S. Robertson, et al., "Okapi at TREC-4", Proceedings of the 4th annual Text Retrieval Conference (TREC-4), NIST, November 1995.

[3] K. Kwok, et al., "TREC-7 Ad-Hoc, High precision and filtering experiments using PIRCS", Proceedings of the 7th annual Text Retrieval Conference (TREC-7), NIST, November 1998.

[4] S. Brin, L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", WWW7 / Computer Networks 30(1-7): 107-117 (1998).

[5] E.A. Fox and J.A. Shaw, “Combination of Multiple Searches,” Proceedings of the 2nd Text Retrieval Conference (TREC-2), NIST Special Publication 500-215, pp. 243-252, 1994.

[6] B.T. Bartell, G.W. Cottrell, and R.K. Belew, “Automatic Combination of multiple ranked retrieval systems,” Proceedings of the 17th Annual ACM-SIGIR, pp. 173-181, 1994.

[7] N.J. Belkin, C. Cool, W.B. Croft and J.P. Callan, “The effect of multiple query representations on information retrieval performance,” Proceedings of the 16th Annual ACM-SIGIR, pp. 339-346, 1993.

[8] N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw, “Combining evidence of multiple query representation for information retrieval,” Information Processing & Management, Vol. 31, No. 3, pp. 431-448, 1995.

[9] J.H. Lee, “Combining Multiple Evidence from Different Properties of Weighting Schemes,” Proceedings of the 18th Annual ACM-SIGIR, pp. 180-188, 1995.

[10] J. Aslam and M. Montague, et al., “Models for Metasearch”, Proceedings of the 24th Annual ACM Conference on Research and Development in Information Retrieval (SIGIR), September 2001.

[11] M. Montague, et al., “Relevance Score Normalization for Metasearch”, Proceedings of the 10th Annual ACM Conference for Information and Knowledge Management (CIKM), 2001.

[12] M. Montague, et al., “Condorcet Fusion for Improved Retrieval”, Proceedings of ACM-CIKM, November 2002.

[13] P. Ogilvie, J. Callan, "Combining Document Representations for Known-Item Search," Proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval (SIGIR), 2003.

[14] A. Chowdhury, et al., "Improved query precision using a unified fusion model", Proceedings of the 9th Text Retrieval Conference (TREC-9), November 2000.

[15] J. Xu, B. Croft, “Corpus-based stemming using co-occurrence of word variants”. ACM Transactions on Information Systems, January, 1998.

[16] Porter, “An algorithm for suffix stripping”. Program, 14(3):130—137, 1980.

[17] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In Proc. of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pages 27{34. Association for Computing Machinery, 2002.

[18] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98) 275-281, 1998.