Experiments in Named Page Finding and Arabic Retrieval with Hummingbird SearchServer^{TM} at TREC 2002

Stephen Tomlinson
Hummingbird
Ottawa, Ontario, Canada
stephen.tomlinson@hummingbird.com
http://www.hummingbird.com/

February 7, 2003

Abstract

Hummingbird participated in the named page finding task of the TREC 2002 Web Track (find the named page in 18GB from the .GOV domain) and the monolingual Arabic topic relevance task of the TREC 2002 Cross-Language Track (find all relevant documents in 869MB of Arabic news data). In the named page finding task, SearchServer returned the named page in the first 10 rows for more than 80% of the 150 queries. Searching the full document content produced mean reciprocal rank (MRR) scores more than 20 points higher than just searching particular HTML properties (such as the Title), but enhancing a content search with a little extra weight for HTML properties further increased MRR by 6 points (with standard error of just 2 points). Treating queries as phrases was not found to help significantly (on average), but document length normalization increased MRR by more than 20 points. For Arabic topic relevance, light algorithmic stemming increased mean average precision (MAP) by 5 points, use of Arabic stop words increased MAP by 1 point, and query expansion from blind feedback increased MAP by 3 points.

1 Introduction

Hummingbird SearchServer[1] is an indexing, search and retrieval engine for embedding in Windows and UNIX information applications. SearchServer, originally a product of Fulcrum Technologies, was acquired by Hummingbird in 1999. Founded in 1983 in Ottawa, Canada, Fulcrum produced the first commercial application program interface (API) for writing information retrieval applications, Fulcrum® Ful/Text^{TM}. The SearchServer kernel is embedded in many Hummingbird products, including SearchServer, an application toolkit used for knowledge-intensive applications that require fast access to unstructured information.

SearchServer supports a variation of the Structured Query Language (SQL), SearchSQL^{TM}, which has extensions for text retrieval. SearchServer conforms to subsets of the Open Database Connectivity (ODBC) interface for C programming language applications and the Java Database Connectivity (JDBC) interface for Java applications. Almost 200 document formats are supported, such as Word, WordPerfect, Excel, PowerPoint, PDF and HTML.

SearchServer works in Unicode internally [5] and supports most of the world's major character sets and languages. The major conferences in text retrieval evaluation (TREC [10], CLEF [2] and NTCIR [7]) have provided opportunities to objectively evaluate SearchServer's support for a dozen languages.

[1] Fulcrum® is a registered trademark, and SearchServer^{TM}, SearchSQL^{TM}, Intuitive Searching^{TM} and Ful/Text^{TM} are trademarks of Hummingbird Ltd. All other copyrights, trademarks and tradenames are the property of their respective owners.

This paper looks at experimental work with SearchServer for named page finding (just one right answer, i.e. a known-item search task) and monolingual Arabic retrieval (find all the relevant documents, i.e. a topic relevance task). All experiments were conducted on a single-cpu desktop system, OTWEBTREC, with a 600MHz Pentium III cpu, 512MB RAM, 186GB of external disk space on one e: partition, running Windows NT 4.0 Service Pack 6. For the submitted runs in July 2002, an internal development build of SearchServer 5.3 was used (5.3.500.264).

2 Named Page Finding

The .GOV collection of the TREC 2002 Web Track consists of pages downloaded from the .gov domain of the World Wide Web in early 2002. It was distributed on 7 CDs. We copied the contents of each CD onto a "compressed NTFS" area of OTWEBTREC's e: drive (e:\data\compressed\gov\cd1 - e:\data\compressed\gov\cd7). The TRANS.TBL files were not considered part of the collection and were removed. The 4613 .gz files comprising the collection were uncompressed (and Windows NT internally recompressed them on the compressed NTFS drive). Uncompressed, the 4613 files consist of 19,455,030,550 bytes (18.1 GB). Based on the change in bytes free on the drive, the files occupied 9,329,721,344 bytes (8.7 GB) on the compressed NTFS drive. Hence, NTFS compression saved about 9.4 GB of space, noticeably less than gzip compression (which saved 13.9 GB). Each file contains on average about 270 "documents", for a total of 1,247,753 documents. The average document size is 15,592 bytes. For more information on the .GOV collection, see [4].

2.1 Indexing

The custom text reader called cTREC, described in last year's paper [13], was enhanced for the named page finding experiments.

In expansion mode, cTREC, which previously just extracted the DOCNO identifier, was enhanced to support a /H option for extracting the following property information from each .GOV document's header and content for storage in columns 129-140 of the SearchServer table:

- 129: "non-empty" title, which was filled with the first non-empty value of columns 130-135 (i.e. the title was used if there was one, otherwise the meta title was used if there was one, etc., with the URL used as a last resort).
- 130: title (text following <TITLE> up to </TITLE>, if any).
- 131: meta title (content of the META TITLE tag, if any).
- 132: meta subject (content of the META SUBJECT tag, if any).
- 133: meta description (content of the META DESCRIPTION tag, if any).
- 134: first heading (text following the first occurrence of the <H1>, <H2> or <H3> tag up to its closing tag, if any).
- 135: URL (which was always included in the DOCHDR section, before the document content).
- 136: URL type (calculated from the URL as described below).
- 137: URL depth (calculated from the URL as described below).
- 138: meta keywords (content of the META KEYWORDS tag, if any).
- 139: all properties except keywords (i.e. a concatenation of columns 130-135).
- 140: all properties (a concatenation of columns 139 and 138).

The URL was truncated at 256 bytes (only 5 were longer), and the other properties were truncated at 1024 bytes.

The URL type was set to ROOT, SUBROOT, PATH or FILE, based on the convention which worked well last year for the Twente/TNO group [14] on the entry page finding task (also known as the home page finding task). Our exact rules were as follows. The slash count of the URL was calculated (a count of the '/' characters not including the leading "http://"). The URL was considered of homepage-type if it ended with "/", "/index.html", "/index.htm", "/default.html", "/default.htm", "/default.asp", "/home.html", "/home.htm", "/welcome.html" or "/welcome.htm" (case-insensitive comparisions were used). ROOT was assigned if the URL was of slash count 0 or was a homepage-type URL of slash count 1. SUBROOT was assigned if the URL was a homepage-type URL of slash count 2. PATH was assigned if the URL was a homepage-type URL of slash count 3 or more. FILE was assigned for all other URLs.

The URL depth was based on the sum of the slash count and the node count, minus one if the URL was of homepage-type. The node count was the count of the dots before the first slash after "http://" (not counting the first dot if the URL began with "http://www.") plus the number of "?", ";" or "#" characters. (As every URL contained ".gov", the URL depth was guaranteed to be at least 1.) For convenience of wildcard searching and readability, the depth was converted to a term as follows: 1 was assigned URLDEPTHA, 2 was assigned URLDEPTHAB, 3 was assigned URLDEPTHABC, etc. with depths greater than 25 treated the same as 25.

In format translation mode, cTREC was enhanced to support a /q option which resumed indexing at the first quotation mark inside a tag, rather than always waiting for the end of the tag to resume indexing. This option hence would index potentially helpful text such as VALUE fields of INPUT tags and NAME fields of IMG tags, although more noise would also be indexed.

For the .GOV collection, the documents were assumed to be in Latin-1, and as for the web collections of past years, the /w option of cTREC was used to convert non-ASCII Latin-1 bytes to the ASCII range (if any occurred).

A SearchServer table called GOV was created for the .GOV collection with the following SearchSQL statement:

```
create schema GOV
create table GOV
(
DOCNO varchar(256) 128,
NONEMPTY_TITLE varchar(2048) 129,
TITLE varchar(2048) 130,
META_TITLE varchar(2048) 131,
META_SUBJECT varchar(2048) 132,
META_DESCRIPTION varchar(2048) 133,
FIRST_HEADING varchar(2048) 134,
URL varchar(2048) 135,
URL_TYPE varchar(2048) 136,
URL_DEPTH varchar(2048) 137,
META_KEYWORDS varchar(2048) 138,
ALL_BUT_KEYWORDS varchar(2048) 139,
ALL_PROPS varchar(2048) 140
)
periodic
stopfile 'mygov.stp'
basepath 'e:\data\compressed';
```

The DOCNO column was assigned number 128 and the remaining columns were assigned numbers 129-140 to correspond to the properties written by the /H option the cTREC text reader. (The reserved external text column, FT_TEXT, which corresponds to the document content, does not need to be specified in the schema.) The mygov.stp stopfile of 99 stop words is a little different from previous years in that it no longer contains single letters or any numbers.

Into the GOV table, just one row was inserted, specifying the top directory of the data set relative to the basepath:

```
insert into GOV ( ft_sfname, ft_flist )
values ( 'gov', 'cTREC/E/d=128/H:s!cTREC/w/q/@:s');
```

To index the GOV table, a Validate Index statement was executed:

```
validate index GOV validate table;
```

2.2 Searching

For the named page finding task of the Web Track, the 150 “topics” were in a file called “web-named_page_topics.1-150.txt”. The topics were numbered NP1-NP150, and each contained a description of a page (e.g. “visiting pandas national zoo”). The task was to rank the named page as highly as possible. The topics were assumed to be in the Latin-1 character set, the default on North American Windows systems (though accent-sensitive searching was not enabled for the GOV table).

For the submitted hum02pd run of the named page finding task, below is an example SearchSQL query. This query would create a working table with the 2 columns named in the SELECT clause, a REL column containing the relevance value of the row for the query, and a DOCNO column containing the document's identifier. The ORDER BY clause specifies that the most relevant rows should be listed first. The statement “SET MAX_SEARCH_ROWS 50” was previously executed so that the working table would contain at most 50 rows:

```
SELECT RELEVANCE('V2:3') AS REL, DOCNO
FROM GOV
WHERE
 (ALL_PROPS CONTAINS 'visiting pandas national zoo' WEIGHT 1) OR
 (ALL_PROPS IS_ABOUT 'visiting pandas national zoo' WEIGHT 1) OR
 (FT_TEXT IS_ABOUT 'visiting pandas national zoo' WEIGHT 10)
ORDER BY REL DESC;
```

The ALL_PROPS column contained all the properties of column 140 (described earlier), e.g. the title and meta description but not most of the document content.

The CONTAINS predicate does phrase searching, so the listed terms would have to occur adjacently in the specified order (except stop words). “SET PHRASE_DISTANCE 4” was previously specified so that there could be up to 4 characters between adjacent terms (plus additional whitespace). By default, the CONTAINS predicate does exact searching (i.e. no stemming), though some normalizations (e.g. case normalization and canonical Unicode) are still done. The motivation for including the query as a phrase was that it seemed the query might often be in the title or other property information of the document (e.g. a query in mind was “Washington State Legislature” (which was not one of the 150 official queries)). The phrase searching was just given one-tenth the weight of content searching for relevance ranking purposes. Experiments on last year's entry page finding task suggested a small weight was helpful (on average) but a strong weight hurt results.

The IS_ABOUT predicate uses SearchServer's Intuitive Searching, described in last year's paper [13]. It by default uses English stemming and just requires one of the terms to match. It was used with WEIGHT 1 on the ALL_PROPS column to increase the ranking of documents with the query in the title or other property information. It was used with WEIGHT 10 on the FT_TEXT column (which represents the external document). Again, these weights were chosen based on what worked well on the previous year's entry page finding task.

For the submitted hum02upd and hum02up runs, a higher weight was given to URLs of particular type and depth, using a SearchSQL WHERE clause of the following form which was was found to work well on last year's entry page finding task:

```
WHERE
((ALL_PROPS CONTAINS 'visiting pandas national zoo' WEIGHT 1) OR
 (ALL_PROPS IS_ABOUT 'visiting pandas national zoo' WEIGHT 1) OR
 (FT_TEXT IS_ABOUT 'visiting pandas national zoo' WEIGHT 10)
) AND (
 (URL_TYPE CONTAINS 'ROOT' WEIGHT 10) OR
 (URL_TYPE CONTAINS 'SUBROOT' WEIGHT 10) OR
 (URL_TYPE CONTAINS 'PATH' WEIGHT 10) OR
 (URL_TYPE CONTAINS 'FILE' WEIGHT 0) OR
 (URL_DEPTH CONTAINS 'URLDEPTHA' WEIGHT 5) OR
 (URL_DEPTH CONTAINS 'URLDEPTHAB' WEIGHT 5) OR
 (URL_DEPTH CONTAINS 'URLDEPTHABC' WEIGHT 5) OR
 (URL_DEPTH CONTAINS 'URLDEPTHABCD' WEIGHT 5) )
```

Although it might seem this query is giving the same weight to the ROOT, SUBROOT and PATH types of URLs (all WEIGHT 10), because a ROOT term is much less frequent in the URL_TYPE column, it in effect gets a higher weight in relevance ranking because of its higher inverse document frequency (and SUBROOT has more impact than PATH for the same reason). The URL type of FILE was given WEIGHT 0, which means it did not affect the relevance calculation, but it was included so that the AND clause would match all rows.

Similarly, giving URL depths 1-4 some extra weight was found to be modestly helpful on last year's entry page finding task. Again, URLs of depth 1 (for which URLDEPTHA was included in the URL_DEPTH column) internally had a higher weight from the inverse document frequency.

For the submitted hum02uhp run, an even higher weight was given to URL_TYPE (the 3 terms of WEIGHT 10 were given WEIGHT 25). On last year's entry page finding task, the stronger URL_TYPE weights gave similar MRR scores to the lower ones.

For the submitted hum02ud run, the SearchSQL query was the same as for hum02upd except that the ALL_PROPS searches were omitted (i.e. properties and phrases in properties were not given extra weight). Note that the FT_TEXT column indexed all of the properties except for the URL of the document header.

The difference between the hum02upd and hum02up runs was in the importance of document length normalization (in general, runs ending with 'd' used "SET RELEVANCE_DLEN_IMP 500" and the others used "SET RELEVANCE_DLEN_IMP 250").

For the named page queries, no query terms were discarded (e.g. there was no expectation that discarding the words "find", "relevant" and "document" would be beneficial, unlike for some previous year's tasks). Of course, the index omitted a few stop words (e.g. "the", "by") as previously mentioned.

For the named page queries, besides linguistic expansion from stemming in the IS_ABOUT predicate, we did not do any query expansion. For example, we did not use approximate text searching for spell-correction (the organizers tried to ensure the topics were spelled correctly), and we did not use row expansion or any other kind of blind feedback technique.

SearchServer's relevance value calculation is the same as described last year [13]. Briefly, SearchServer dampens the term frequency and adjusts for document length in a manner similar to Okapi [8] and dampens the inverse document frequency using an approximation of the logarithm. SearchServer's relevance values are always an integer in the range 0 to 1000.

When multiple predicates are combined, as was done for the named page queries this year, SearchServer currently does not normalize by query length. For example, the URL_TYPE clauses of the earlier examples would have a lot less relative impact if the named page query contained 5 words instead of 1.

SearchServer's RELEVANCE_METHOD setting can be used to optionally square the importance of the inverse document frequency (by choosing a RELEVANCE_METHOD of 'V2:4' instead of 'V2:3'). The importance of document length to the ranking is controlled by SearchServer's RELEVANCE_DLEN_IMP setting (scale of 0 to 1000).

Table 1: Scores of Submitted Named Page Finding Runs

Run	MRR	%Top10	%Fail
hum02pd	0.626	82.0%	9.3%
hum02upd	0.538	75.3%	13.3%
hum02up	0.527	74.0%	11.3%
hum02ud	0.456	68.0%	16.7%
hum02uhp	0.337	51.3%	33.3%

Table 2: Impact of Submitted Named Page Finding Techniques on Reciprocal Rank

Experiment	AvgDiff	95% Confidence	vs.	2 Largest Diffs (Topic)
p (upd - ud)	0.082	(0.041, 0.124)	54-17-79	0.900 (2), 0.889 (51)
d (upd - up)	0.011	(−0.020, 0.043)	33-28-89	0.833 (36), 0.800 (41)
u (upd - pd)	−0.088	(−0.130,−0.047)	15-50-85	−1.000 (64), −0.917 (138)
h (uhp - up)	−0.190	(−0.236,−0.146)	0-87-63	−1.000 (2), −1.000 (117)

2.3 Official Results

The evaluation measures are likely explained in an appendix of this volume. Briefly, "Reciprocal Rank" for a topic is one divided by the rank in which the named page was found (using the smallest rank if there were duplicates of the named page), or zero if the named page was not found. "Mean Reciprocal Rank" (MRR) is the average of the reciprocal ranks over all the topics. "%Top10" is the percentage of topics for which the named page was found in the first 10 rows. "%Fail" is the percentage of topics for which the named page was not found in the first 50 rows.

Table 1 shows the scores of the submitted named page finding runs.

Most of the remaining tables will focus on one particular precision measure (usually reciprocal rank or average precision), comparing the scores when a particular feature (such as stemming) is enabled to when it is disabled. The columns of these tables are as follows:

- "Experiment" is the feature tested.
- "AvgDiff" is the average difference in the score.
- "95% Confidence" is an approximate 95% confidence interval for the average difference calculated using Efron's bootstrap percentile method[2] [3] (using 100,000 iterations). If zero is not in the interval, the result is "statistically significant" (at the 5% level), i.e. the feature is unlikely to be of neutral impact, though if the average difference is small (e.g. <0.020) it may still be too minor to be considered "significant" in the magnitude sense.
- "vs." is the number of topics on which the score was higher, lower and tied (respectively) with the feature enabled. These numbers should always add to the number of topics for the task.
- "2 Largest Diffs (Topic)" lists the two largest differences in the precision score (based on the absolute value), with each followed by the corresponding topic number in brackets (the named page topic numbers range from 1 to 150).

Table 2 shows the impact when isolating each technique distinguishing the submitted named page finding runs:

[2] See [12] for some comparisons of confidence intervals from the bootstrap percentile, Wilcoxon signed rank and standard error methods for both average precision and Precision@10.

- The ‘p’ factor (extra weight for HTML properties and phrases in properties) increased MRR 8 points. Some diagnostics of this result, including whether it holds up when URL techniques are not also used, are in the next section.
- The ‘d’ factor (document length importance of 500 instead of 250) made little difference.
- The ‘u’ factor (extra weight for URL type and depth) lowered MRR by 9 points, contrary to its substantial beneficial impact on last year’s entry page task.
- The ‘h’ factor (even more extra weight for URL type) lowered MRR by another 19 points, even though it had a neutral impact on last year’s entry page task.

2.4 Diagnostic Results

For the diagnostics, we defined a “base run” which set the document length importance to 500 and executed an IS_ABOUT search of just the content (i.e. the FT_TEXT column). For example:

```
SELECT RELEVANCE('V2:3') AS REL, DOCNO
FROM GOV
WHERE (FT_TEXT IS_ABOUT 'visiting pandas national zoo')
ORDER BY REL DESC;
```

The base run scored a 0.564 mean reciprocal rank, finding the named page in the top 10 for 75.3% of the queries while failing to find it in the top 50 for 11.3% of the queries.

Table 3 shows a comparison of various runs to the base run (always subtracting the base run’s scores in reciprocal rank from the listed run). The first row compares submitted run hum02pd to the base run, like the above ‘p’ factor experiment but without the ill-fated URL techniques. While the average gain from properties and phrases is a little smaller (6 points), the (approximate) 95% confidence interval (1 to 11 points) indicates it is still statistically significant. Topic 64 (work/life center map) benefitted most.

The runs in the next group of comparisons in Table 3 (listed with a leading + sign) added the listed column to the WHERE clause with one-tenth the weight of the content search. For example, for the “+ TITLE” row, the query’s WHERE clause was of this form:

```
WHERE
 (TITLE   IS_ABOUT 'visiting pandas national zoo' WEIGHT 1 ) OR
 (FT_TEXT IS_ABOUT 'visiting pandas national zoo' WEIGHT 10)
```

Apparently it is the extra weight on particular columns and not the use of phrases which explains most of the gain of the ‘p’ factor. For example, the “+ ALL_PROPS” run differs from hum02pd in that phrases are not used, but it produces similar gains. Of the HTML properties, just giving extra weight to the TITLE produces most of the gains. The URL and FIRST_HEADING also appear to be helpful for named page finding on average, while the META properties were harmful on average, but most of these latter results were not statistically significant. Note that the FT_TEXT column included the content of the other columns except for the URL column.

The next group of rows in Table 3 shows the importance of the content to named page finding. The compared runs just searched the listed column. For example, for the “TITLE” row, the query’s WHERE clause was of this form:

```
WHERE (TITLE IS_ABOUT 'visiting pandas national zoo')
```

The content column (FT_TEXT) scored significantly higher, on average, than any other column by itself. The end of the confidence interval closest to zero represents a difference of at least 13 points in every case. Still, the “vs.” column shows that for approximately one-sixth of the queries, just searching the columns containing the TITLE outscored content searching.

The last group of rows contains some miscellaneous experiments. The results with positive impacts were not statistically significant, but the negative impacts were. Just searching for the named page query as a

Table 3: Comparison with Plain Content Diagnostic Run in Reciprocal Rank

Experiment	AvgDiff	95% Confidence	vs.	2 Largest Diffs (Topic)
hum02pd	0.061	(0.017, 0.107)	49-16-85	0.968 (64), 0.909 (51)
+ ALL_BUT_KEYWORDS	0.058	(0.014, 0.103)	46-15-89	0.968 (64), 0.909 (51)
+ ALL_PROPS	0.055	(0.013, 0.097)	48-15-87	0.968 (64), 0.909 (51)
+ NONEMPTY_TITLE	0.047	(0.008, 0.086)	43-15-92	0.909 (51), 0.900 (38)
+ TITLE	0.042	(0.004, 0.080)	41-16-93	0.909 (51), 0.900 (38)
+ URL	0.034	(0.005, 0.066)	26-20-104	0.909 (51), 0.900 (106)
+ FIRST_HEADING	0.034	(−0.001, 0.071)	25-26-99	0.909 (51), 0.889 (138)
+ META_TITLE	−0.006	(−0.022, 0.012)	10-14-126	0.857 (143), −0.500 (101)
+ META_SUBJECT	−0.012	(−0.025,−0.001)	7-15-128	−0.500 (101), −0.500 (57)
+ META_DESCRIPTION	−0.022	(−0.052, 0.009)	19-31-100	0.889 (2), −0.857 (61)
+ META_KEYWORDS	−0.024	(−0.052, 0.002)	20-30-100	−0.750 (61), −0.667 (93)
FT_TEXT	0.000	(−0.001, 0.001)	0-0-150	0.000 (76), 0.000 (2)
ALL_PROPS	−0.212	(−0.292,−0.131)	25-81-44	−1.000 (124), −1.000 (69)
ALL_BUT_KEYWORDS	−0.218	(−0.299,−0.138)	29-79-42	−1.000 (108), −1.000 (28)
NONEMPTY_TITLE	−0.277	(−0.357,−0.196)	23-87-40	−1.000 (132), −1.000 (117)
TITLE	−0.268	(−0.347,−0.188)	23-86-41	−1.000 (92), −1.000 (107)
FIRST_HEADING	−0.445	(−0.524,−0.363)	14-114-22	−1.000 (61), −1.000 (35)
META_DESCRIPTION	−0.487	(−0.563,−0.409)	8-119-23	−1.000 (90), −1.000 (27)
URL	−0.499	(−0.577,−0.420)	10-122-18	−1.000 (92), −1.000 (43)
META_KEYWORDS	−0.502	(−0.574,−0.430)	4-123-23	−1.000 (84), −1.000 (85)
META_TITLE	−0.541	(−0.612,−0.470)	2-128-20	−1.000 (77), −1.000 (80)
META_SUBJECT	−0.562	(−0.631,−0.491)	1-132-17	−1.000 (1), −1.000 (80)
+ phrase	0.015	(−0.012, 0.045)	18-11-121	0.950 (68), 0.909 (51)
stemming off	0.013	(−0.014, 0.041)	32-16-102	−0.957 (150), 0.900 (38)
DLEN 750	0.011	(−0.016, 0.039)	30-20-100	−0.800 (85), 0.750 (139)
idf squared	−0.025	(−0.049,−0.001)	16-32-102	−0.833 (96), −0.667 (84)
DLEN 0	−0.210	(−0.262,−0.158)	7-85-58	−1.000 (36), −0.974 (96)
phrase only	−0.424	(−0.503,−0.345)	11-108-31	−1.000 (97), −1.000 (92)

phrase in the document (WHERE FT_TEXT CONTAINS ’query’), as in the “phrase only” row, significantly hurt on average, but enhancing the base run by giving a little extra weight (one-tenth) to the query as a phrase (OR FT_TEXT contains ’query’), as in the “+ phrase” row, was modestly helpful. Disabling stemming (via SET VECTOR_GENERATOR ‘’) was only modestly helpful as per the “stemming off” row. Increasing the importance of document length normalization (via SET RELEVANCE_DLEN_IMP 750, as opposed to the base run setting of 500) didn’t make much difference (as per the “DLEN 750” row), but decreasing it to 0 (as per the “DLEN 0”) row significantly hurt. Increasing the importance of inverse document frequency (by using relevance method ‘V2:4’ instead of ‘V2:3’) was modestly detrimental as per the “idf squared” row. Even for the impacts which were modest on average, individual queries could have large changes in their scores as indicated by the “2 Largest Diffs” column.

3 Arabic Retrieval

The Arabic document set was the same as last year: Arabic Newswire A Corpus [1] consisting of articles from the Agence France Presse (AFP) Arabic Newswire from 1994-2000. It contained 383,872 documents, totalling 911,555,745 bytes (869 MB) uncompressed.

Table 4: Impact of Submitted Arabic Techniques on Average Precision

Experiment	AvgDiff	95% Confidence	vs.	2 Largest Diffs (Topic)
Exp (tde - td)	0.033	(0.021, 0.046)	37-13-0	0.204 (27), 0.131 (58)
Morph+Stop (td - tdm)	0.032	(0.005, 0.061)	34-16-0	0.360 (29), 0.335 (60)
Narr (tdne - tde)	0.027	(−0.014, 0.073)	27-22-1	0.794 (34), −0.551 (27)
Desc (tde - te)	0.014	(−0.011, 0.039)	29-20-1	0.348 (58), −0.311 (49)

3.1 Indexing

SearchServer (as of version 5.0) internally uses the Unicode canonical decomposition of text and, by default, does not index combining characters (accents, diacritics, etc.). For Arabic, this means by default that composite characters 0622 (alef with madda above), 0623 (alef with hamza above) and 0625 (alef with hamza below) are treated as 0627 (alef), 0624 (waw with hamza above) becomes 0648 (waw), and 0626 (yeh with hamza above) becomes 064A (yeh) (the codes and names are from the Unicode Standard [15]). All of our submitted runs both last year and this year used this default behaviour.

For the submitted runs, two different SearchServer tables called ARAB01 and ARAB01AS were created. ARAB01 was the same as last year, i.e. no stop words and no Arabic morphological normalizations were used. ARAB01AS used the stop words [9] and the experimental morphological normalizations described in section 5.2 of last year's paper [13] (which were just used for diagnostic runs last year, not submitted runs like this year).

3.2 Searching

Compared to last year, there were twice as many Arabic topics (50). They were numbered AR26-AR75 and were distributed in a file called "final_arabic02.txt". The topics contained a "Title" (subject of the topic), "Description" (typically a one-sentence specification of the information need) and "Narrative" (more detailed guidelines for what a relevant document should or should not contain). The topics were encoded in the ISO 8859-6 character set, so "SET CHARACTER_SET 'ISO_8859_6'" was executed before the searches.

Like last year, Intuitive Searching of the content was used (i.e. FT_TEXT IS_ABOUT). The statement "SET MAX_SEARCH_ROWS 1000" was previously executed so that the working table would contain at most 1000 rows. There were no experiments with phrases nor columns other than FT_TEXT.

Submitted runs humAR02tdm and humAR02td both used the Title and Description fields in the query. Run humAR02tdm searched the ARAB01 table, and run humAR02td searched the ARAB01AS table. The other settings (e.g. RELEVANCE_METHOD 'V2:3' and RELEVANCE_DLEN_IMP 500) were the same as described in section 5.2 of the final version of last year's paper [13].

Submitted run humAR02tde used query expansion from blind feedback in the same way as described in last year's paper [13]. The base run was humAR02td and the first 5 rows were used to generate broader queries.

Submitted runs humAR02te and humAR02tdne differed from humAR02tde in that the former just used the Title field in its base run, and the latter additionally used the Narrative field in its base run.

3.3 Official Results

To review the evaluation measures for topic relevance tasks: "Precision" is the percentage of retrieved documents which are relevant. "Precision@n" is the precision after n documents have been retrieved. "Average precision" for a topic is the average of the precision after each relevant document is retrieved (using zero as the precision for relevant documents which are not retrieved). "Recall" is the percentage of relevant documents which have been retrieved. "Interpolated precision" at a particular recall level for a topic is the maximum precision achieved for the topic at that or any higher recall level. For a set of topics, the measure is the average of the measure for each topic (i.e. all topics are weighted equally).

Table 5: Precision of Arabic Diagnostic Runs

Run	AvgP	P@5	P@10	P@20	Rec0	Rec30	P@R
C-td-Y01	0.180	46.4%	42.8%	41.4%	0.697	0.232	23.7%
def-td-Y01	0.209	48.0%	48.4%	46.2%	0.724	0.285	27.1%
L-td-Y01	0.286	61.6%	54.0%	49.0%	0.803	0.379	34.7%
CL-td-Y01	0.302	64.0%	56.8%	51.2%	0.833	0.402	36.2%
CLS-td-Y01	0.337	64.8%	58.4%	52.0%	0.851	0.444	38.0%
CLSE-td-Y01	0.365	64.8%	58.4%	52.8%	0.836	0.477	40.3%
C-td-Y02	0.224	31.6%	34.8%	31.6%	0.595	0.302	26.6%
def-td-Y02	0.227	32.0%	35.8%	32.5%	0.589	0.310	27.4%
L-td-Y02	0.273	42.8%	38.8%	37.1%	0.651	0.354	30.5%
CL-td-Y02	0.278	42.8%	38.4%	36.5%	0.687	0.361	31.2%
CLS-td-Y02	0.291	43.2%	39.8%	36.4%	0.712	0.374	31.6%
CLSE-td-Y02	0.323	44.8%	43.6%	39.5%	0.707	0.426	34.5%

The scores of the submitted runs are expected to be listed in the appendix of the conference proceedings. Table 4 shows the impact when isolating each technique distinguishing the submitted runs. The query expansion technique ("Exp" experiment) increased mean average precision by 3 points and was fairly consistent (small standard error), as evidenced by the narrow confidence interval. The experimental morphological normalizations plus stop words ("Morph+Stop") also increased mean average precision by 3 points, but less consistently, as evidenced by the wider confidence interval. Including the Narrative field ("Narr") increased mean average precision by 3 points, but was very inconsistent; using the Narrative often hurt the scores. Including the Description field ("Desc") increased mean average precision by just 1 point, though again was not very consistent.

3.4 Diagnostic Results

After the official runs were submitted, we used SearchServer's plug-in parser architecture to experiment with plugging in an implementation of the light algorithmic "Light8" stemmer of Larkey et al. [6]. It contained several stemming rules we were not previously using. On topic AR30, we found the light stemmer, in combination with the default dropping of combining characters, did not stem Arabic words for "satellite" to the same form, leading us to also experiment with indexing combining characters 0653 (maddah above), 0654 (hamza above) and 0655 (hamza below) via an extra line in the stopfile ("IAC="\u0653-\u0655" "). But the light stemmer explicitly dropped these combining characters if they followed 0627 (alef).

Table 5 lists precision scores for the diagnostic runs. Listed for each run are its mean average precision (AvgP), the mean precision after 5, 10 and 20 documents retrieved (P@5, P@10 and P@20 respectively), the mean interpolated precision at 0% and 30% recall (Rec0 and Rec30 respectively), and the mean precision after R documents retrieved (P@R) where R is the number of relevant documents for the topic. The following run codes were used: "Y01" (Year 2001) specifies the run used the 25 TREC 2001 topics. "Y02" (Year 2002) specifies the run used the 50 TREC 2002 topics. "L" (Light stemming) specifies the run used the light algorithmic stemmer. "C" (Combining characters) specifies that combining characters 0653-0655 were not dropped by SearchServer. "S" (Stop words) specifies the run used a table which did not index stop words. "E" (Expansion) specifies the run used query expansion from blind feedback. "td" specifies the Title and Description fields were used. "def" specifies the default settings. In particular, the "def-td-Y02" run of Table 5 is the same as submitted run "humAR02tdm" (a baseline Title+Description run not using light stemming, combining character indexing, stop words nor expansion).

Table 6 isolates the impact of various techniques on the average precision measure. All of these comparisons use "td" topics, and most of them are statistically significant at 5% level:

- The "+L" rows isolate the impact of light stemming. The impact when indexing combining characters

Table 6: Impact of Diagnostic Arabic Techniques on Average Precision

Experiment	AvgDiff	95% Confidence	vs.	2 Largest Diffs (Topic)
+L (L-def) td-Y01	0.078	(0.027, 0.134)	17-8-0	0.424 (19), 0.408 (14)
+C (CL-L) td-Y01	0.016	(0.000, 0.038)	8-5-12	0.231 (16), 0.100 (2)
+LC (CL-def) td-Y01	0.093	(0.045, 0.147)	19-6-0	0.424 (19), 0.408 (14)
+C (C-def) td-Y01	−0.029	(−0.071, 0.000)	6-9-10	−0.423 (10), −0.219 (14)
+L (CL-C) td-Y01	0.122	(0.061, 0.192)	19-6-0	0.627 (14), 0.424 (19)
+S (CLS-CL) td-Y01	0.035	(0.017, 0.056)	22-3-0	0.204 (17), 0.116 (7)
+CLS (CLS-def) td-Y01	0.128	(0.075, 0.188)	21-4-0	0.518 (19), 0.461 (14)
+E (CLSE-CLS) td-Y01	0.029	(0.009, 0.049)	17-8-0	0.138 (15), 0.116 (2)
+CLSE (CLSE-def) td-Y01	0.157	(0.102, 0.217)	23-2-0	0.559 (19), 0.377 (14)
+L (L-def) td-Y02	0.046	(0.016, 0.079)	31-19-0	0.360 (29), 0.358 (60)
+C (CL-L) td-Y02	0.005	(−0.007, 0.023)	18-14-18	0.371 (30), −0.088 (57)
+LC (CL-def) td-Y02	0.051	(0.017, 0.088)	29-21-0	0.479 (30), 0.360 (29)
+C (C-def) td-Y02	−0.003	(−0.012, 0.004)	16-14-20	−0.179 (55), 0.053 (37)
+L (CL-C) td-Y02	0.054	(0.022, 0.089)	31-19-0	0.479 (30), 0.367 (60)
+S (CLS-CL) td-Y02	0.014	(0.007, 0.020)	43-7-0	0.068 (30), 0.056 (45)
+CLS (CLS-def) td-Y02	0.064	(0.031, 0.102)	36-14-0	0.546 (30), 0.363 (60)
+E (CLSE-CLS) td-Y02	0.032	(0.016, 0.049)	35-15-0	0.294 (27), 0.147 (58)
+CLSE (CLSE-def) td-Y02	0.096	(0.057, 0.138)	38-12-0	0.599 (30), 0.424 (27)

("CL-C", i.e. subtracting the "C" run from the "CL" run) was particularly substantial on the TREC 2001 topics, where we found an increase of 0.122 (from .180 to .302). Larkey's comparable figure was 0.182 (from 0.194 to 0.376) which is inside our approximate 95% confidence interval of (0.061, 0.192). For the TREC 2002 topics, which did not exist when the stemmer was developed, we find a smaller, though still significant, increase.

- The "+C" rows isolate the impact of indexing the combining characters. When this was the only change from the default ("C-def"), the impact tended to be detrimental, perhaps because the alef forms were no longer conflated. When applied to light stemming runs ("CL-L"), the light stemmer re-conflated the alefs, and the net impact (in effect preserving composite characters 0624 and 0626) tended to be beneficial.

- The "+LC" rows show the combined impact of light stemming and indexing combining characters. Of course, the average impacts add within rounding differences (the calculations were done to 4 decimal places though just 3 are shown). However, confidence interval endpoints do not add (e.g. 0.027 plus 0.000 does not add to 0.045).

- The "+S" rows isolate the impact of using the Arabic stop word list (subtracting the "CL" run from the "CLS" run). The increases are small but fairly consistent, much like in European stop word experiments when using the full topics [12], but for the Arabic task this result also holds when omitting the Narrative.

- The "+E" rows isolate the impact of the query expansion technique (subtracting the "CLS" run from the "CLSE" run). The results are similar to the official "Exp" experiment. As the expansion terms are chosen from the first 5 rows, the first 5 rows are usually the same after expansion, which moderates how much the result can change. We haven't done a lot of work on our expansion technique and it is likely underachieving. Expanding queries generally leads to much longer processing times which can be a high price to pay for improvements in the part of the result list that users might not even look at. In practical systems, users can control the query terms themselves rather than depend on blind feedback.

- The “+CLSE” rows show the combined impact of all 4 techniques. Even the low end of the confidence intervals represent a substantial impact.

In all 9 cases in Table 6, the confidence interval for the 50 topic experiment (Y02) was narrower than the confidence interval for the corresponding 25 topic experiment (Y01). Also reassuringly, the corresponding confidence intervals always overlapped. The interval widths ranged from 1 to 8 points for the 50 topic experiments and from 4 to 13 points for the 25 topic experiments.

References

[1] Arabic Newswire Part 1, Linguistic Data Consortium (LDC) catalog number LDC2001T55, ISBN 1-58563-190-6. http://www.ldc.upenn.edu/Catalog/LDC2001T55.html

[2] Cross-Language Evaluation Forum web site. http://www.clef-campaign.org/

[3] Bradley Efron and Robert J. Tibshirani. An Introduction to the Bootstrap. 1993. Chapman & Hall/CRC.

[4] The .GOV Test Collection. http://www.ted.cmis.csiro.au/TRECWeb/govinfo.html

[5] Andrew Hodgson. Converting the Fulcrum Search Engine to Unicode. In Sixteenth International Unicode Conference, Amsterdam, The Netherlands, March 2000.

[6] Leah S. Larkey, Lisa Ballesteros and Margaret E. Connell. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng and Kalervo Järvelin, editors, Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 275-282, 2002.

[7] NTCIR (NII-NACSIS Test Collection for IR Systems) Home Page. http://research.nii.ac.jp/∼ntcadm/index-en.html

[8] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford. (City University.) Okapi at TREC-3. In D. K. Harman, editor, Overview of the Third Text REtrieval Conference (TREC-3). NIST Special Publication 500-226. http://trec.nist.gov/pubs/trec3/t3_proceedings.html

[9] Bonnie Glover Stalls and Yaser Al-Onaizan. (University of Southern California, Information Sciences Institute, Natural Language Group.) Arabic Stop Words List. http://www.isi.edu/∼yaser/arabic/arabic-stop-words.html

[10] Text REtrieval Conference (TREC) Home Page. http://trec.nist.gov/

[11] Stephen Tomlinson and Tom Blackwell. Hummingbird's Fulcrum SearchServer at TREC-9. In E. M. Voorhees and D. K. Harman, editors, Proceedings of the Ninth Text REtrieval Conference (TREC-9). NIST Special Publication 500-249. http://trec.nist.gov/pubs/trec9/t9_proceedings.html

[12] Stephen Tomlinson. Experiments in 8 European Languages with Hummingbird SearchServer^{TM} at CLEF 2002. In Carol Peters, editor, Working Notes for the CLEF 2002 Workshop. http://clef.iei.pi.cnr.it:2002/workshop2002/WN/26.pdf

[13] Stephen Tomlinson. Hummingbird SearchServer^{TM} at TREC 2001. In E. M. Voorhees and D. K. Harman, editors, Proceedings of the Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250. http://trec.nist.gov/pubs/trec10/t10_proceedings.html

[14] Thijs Westerveld, Wessel Kraaij and Djoerd Hiemstra. Retrieving Web Pages using Content, Links, URLs and Anchors. In E. M. Voorhees and D. K. Harman, editors, Proceedings of the Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250. http://trec.nist.gov/pubs/trec10/t10_proceedings.html

[15] The Unicode Standard Version 3.0. The Unicode Consortium. 2000. Addison-Wesley.