Hummingbird SearchServer™ at TREC 2001

Stephen Tomlinson[1]
Hummingbird
Ottawa, Ontario, Canada

February 4, 2002

Abstract

Hummingbird submitted ranked result sets for the topic relevance task of the TREC 2001 Web Track (10GB of web data) and for the monolingual Arabic task of the TREC 2001 Cross-Language Track (869MB of Arabic news data). SearchServer's Intuitive Searching™ tied or exceeded the median Precision@10 score in 46 of the 50 web queries. For the web queries, enabling SearchServer's document length normalization increased Precision@10 by 65% and average precision by 55%. SearchServer's option to square the importance of inverse document frequency (V2:4 vs. V2:3) increased Precision@10 by 8% and average precision by 12%. SearchServer's stemming increased Precision@10 by 5% and average precision by 13%. For the Arabic queries, a combination of experimental Arabic morphological normalizations, Arabic stop words and pseudo-relevance feedback increased average precision by 53% and Precision@10 by 9%.

1 Introduction

Hummingbird SearchServer[2] 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™. 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), called SearchSQL™, 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. Many character sets and languages are supported. SearchServer's Intuitive Searching algorithms were updated for version 4.0 which shipped in Fall 1999, and in subsequent releases of other products. SearchServer 5.0, which shipped in Spring 2001, works in Unicode internally [4] and contains improved natural language processing technology, particularly for languages with many compound words, such as German, Dutch and Finnish.

2 System Description

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, and running Windows NT 4.0 Service Pack 6. An

---

[1] Core Technology, Research and Development, stephen.tomlinson@hummingbird.com

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

internal development build of SearchServer 5.0 was used for the official TREC runs in July 2001 (build 5.0.504.156), which for the web and Arabic tasks should give essentially the same rankings as the commercial release version.

3 Setup

We describe how SearchServer was used to handle the topic relevance task of the TREC 2001 Web Track (10GB of web data) and the monolingual Arabic task of the TREC 2001 Cross-Language Track (869MB of Arabic news data).

3.1 Data

The WT10g collection of the Web Track consists of pages downloaded from the World Wide Web in 1997. It was distributed on 5 CDs. We copied the contents of each CD onto the OTWEBTREC e: drive (e:\data\wt10g\cd1 - e:\data\wt10g\cd5). The cd5\info subdirectory, containing supporting information not considered part of WT10g, was removed to ensure it wasn't indexed. The 5157 .gz files comprising WT10g were uncompressed. No further pre-processing was done on the data. Uncompressed, the 5157 files consist of 11,032,691,403 bytes (10.3GB), about 2MB each. Each file contains on average 328 "documents", for a total of 1,692,096 documents. The average document size is 6520 bytes. For more information on this collection, see [2].

Arabic Newswire A Corpus of the Cross-Language Track consists of articles from the Agence France Presse (AFP) Arabic Newswire from 1994-2000. It was distributed on 1 CD. We copied the contents of its TRANSCRIPTS directory to e:\data\Arabic. The 2337 gz files comprising the corpus were uncompressed. No further pre-processing was done on the data. Uncompressed the 2337 files consist of 911,555,745 bytes (869 MB), about 370KB each. Each file contains on average 164 "documents", for a total of 383,872 documents. The average document size is 2375 bytes. For more information on this collection, see [1].

3.2 Text Reader

To index and retrieve data, SearchServer requires the data to be in Fulcrum Technologies Document Format (FTDF). SearchServer includes "text readers" for converting most popular formats (e.g. Word, WordPerfect, etc.) to FTDF. A special class of text readers, "expansion" text readers, can insert a row into a SearchServer table for each logical document inside a container, such as directory or library file. Users can also write their own text readers in C for expanding proprietary container formats and converting proprietary data formats to FTDF.

Last year, for TREC-9, we wrote a custom text reader called cTREC to handle expansion of the library files of WT10g collection and to make a few conversions to the HTML format, described in [10]. We used cTREC again this year and made no significant changes regarding WT10g. This year, we will just describe how cTREC was extended for the Arabic collection.

The library files of the Arabic collection, like WT10g, consist of several logical documents, each starting with a <DOC> tag and ending with a </DOC> tag. After the <DOC> tag, the unique id of the document, e.g. 19940513_AFP_ARB.0001, is included inside <DOCNO>..</DOCNO> tags. The cTREC /E switch handles expansion of the Arabic library files into logical documents identically as for WT10g.

The Arabic documents contain SGML tags describing its structure (e.g. the headline is preceded by a <HEADLINE> tag and followed by a </HEADLINE> tag). The Document Type Definition (DTD) which specified the tags and entities used in the documents was provided in the ldc_arabic.dtd file on the CD. When invoked without the /w or /E switch, cTREC by default inserts control sequences to turn off indexing around all tags listed in the Arabic collection DTD (and additionally tags listed in the TREC disk 1-5 DTDs, as described last year), and converts all entities listed in the DTDs (e.g. "&" is converted to the ampersand "&"). By default, cTREC also

turns off indexing for data delineated by certain tags because its content isn't considered helpful (for the Arabic collection, data delineated by HEADER, FOOTER and TRAILER tags is not indexed). cTREC looks ahead at most 5000 bytes for an end tag when it encounters a tag indicating indexing should be turned off; if the end tag is not found, indexing is not turned off.

The Arabic documents are in the UTF-8 character set, a variable-width encoding of Unicode, for which ASCII characters are represented with 1 byte (e.g. the Latin letter A, which is hexadecimal value 0x0041 in the UTF-16 encoding of Unicode, is 1 byte in UTF-8 (hexadecimal 0x41 or decimal 65)), and non-ASCII characters are represented with 2 to 4 bytes (e.g. the Arabic letter ALEF, which is 0x0627 in UTF-16, is 2 bytes in UTF-8 (0xd8 0xa7)). cTREC passes through the bytes of the documents unchanged (aside from the control sequences inserted and entities converted as described previously). SearchServer's Translation Text Reader (nti), was chained on top of cTREC and the UTF8_UCS2 translation was specified via its /t option to translate from UTF-8 to the UTF-16 encoding desired by SearchServer's parser.

3.3 Indexing

We created a SearchServer table called WT10GW for the web collection and two different SearchServer tables called ARAB01 and ARAB01A for the Arabic collection. For example, the SearchSQL statement to create the ARAB01A table was as follows:

```
CREATE SCHEMA ARAB01A CREATE TABLE ARAB01A
(DOCNO VARCHAR(256) 128) PERIODIC
BASEPATH 'E:\DATA' STOPFILE 'MYARAB.STP';
```

The stopfile differed for each table. For WT10GW, we used the same MYTREC.STP stopfile as last year, which contained 101 stopwords to not index, including all letters and single-digit numbers. For ARAB01, we did not use a stopfile. For ARAB01A, the stopfile MYARAB.STP did not actually contain any stopwords, but specified a non-default option to the parser to apply experimental Arabic morphological normalizations to the words before indexing:

```
PARSER="unicode/a=1"
```

The PARSER line of the stopfile specified the built-in unicode parser with the non-default option of a=1 which enables the experimental Arabic morphological normalizations. A powerful new feature of SearchServer 5.0 is the ability to "plug-in" a custom parser to extend or replace the default parser.

Into each table, we just inserted one row, specifying the top directory of the data set. e.g. for the ARAB01A table, we used this Insert statement:

```
INSERT INTO ARAB01A ( FT_SFNAME, FT_FLIST )
VALUES ( 'ARABIC', 'cTREC/E/d=128:s!nti/t=UTF8_UCS2:cTREC/@:s');
```

To index each table, we just executed a Validate Index statement such as the following:

```
VALIDATE INDEX ARAB01A VALIDATE TABLE
TEMP_FILE_SIZE 2000000000 BUFFER 256000000;
```

The VALIDATE TABLE option of the VALIDATE INDEX statement causes SearchServer to review whether the contents of container rows, such as directory rows and library files, are correctly reflected in the table. In this particular case, SearchServer initially validated the directory row by inserting each of its sub-directories and files into the table. Then SearchServer validated each of those directory and library file rows in turn, etc. Validating

library file rows invoked the cTREC text reader in expansion mode to insert a row for each logical document in the library file, including its document id.

After validating the table, SearchServer indexed the table, in this case using up to 256MB of memory for sorting (as per the BUFFER parameter) and using temporary sort files of up to 2GB (as per the TEMP_FILE_SIZE parameter). The index includes a dictionary of the distinct words (after some Unicode-based normalizations, such as converting to upper-case and decomposed form, and in the case of the ARAB01A table, Arabic-specific normalizations as previously described) and a reference file with the locations of the word occurrences. Additionally, by default, each distinct word is stemmed and enough information saved so that SearchServer can efficiently find all occurrences of any word which has a particular stem. By default, the stemming is done with an English lexicon which has no effect on Arabic words.

4 Search Techniques

For the topic relevance task of the Web Track, the 50 "topics" were in a file called "topics.501-550". The topics were numbered from 501-550, and each contained a Title (which was an actual web query taken from a search engine log), a Description (NIST's interpretation of the query), and a Narrative (a more detailed set of guidelines for what a relevant document should or should not contain). We assumed the topics were in the Latin-1 character set, the default on North American Windows systems.

For the Cross-Language Track, the 25 topics were in 3 different languages (English, French and Arabic). We just used the Arabic topics in a file called "arabic_topics.txt". The Arabic topics were numbered AR1 to AR25. They were encoded in the ISO 8859-6 (Latin-6) character set.

We created an ODBC application, called QueryToRankings.c, based on the example stsample.c program included with SearchServer, to parse the topics files, construct and execute corresponding SearchSQL queries, fetch the top 1000 rows, and write out the rows in the results format requested by NIST. SELECT statements were issued with the SQLExecDirect api call. Fetches were done with SQLFetch (typically 1000 SQLFetch calls per query).

4.1 Character Set

SearchServer easily handled the issue of the Arabic queries and documents being in different character sets. Before running the Arabic queries, the SearchSQL statement "SET CHARACTER_SET 'ISO_8859_6'" was executed so that SearchServer would transcode the queries from Latin-6 to Unicode. The web queries were assumed to be in the Latin-1 character set, the default.

4.2 Intuitive Searching

For all runs, we used SearchServer's Intuitive Searching, i.e. the IS_ABOUT predicate of SearchSQL, which accepts unstructured text. For example, for topic 507 of the Web Track, the Title was "dodge recalls?". A corresponding SearchSQL query would be:

```
SELECT RELEVANCE('V2:3') AS REL, DOCNO
FROM WT10GW
WHERE FT_TEXT IS_ABOUT 'dodge recalls?'
ORDER BY REL DESC;
```

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 1000" was previously executed so that the working table would contain at most 1000 rows.

4.3 Stemming

SearchServer "stems" each distinct word to one or more base forms, called stems, using lexicon-based natural language processing technology. For example, in English, "baby", "babied", "babies", "baby's" and "babying" all have "baby" as a stem. Compound words in languages such as German, Dutch and Finnish produce multiple stems; e.g., in German, "babykost" has "baby" and "kost" as stems.

By default, Intuitive Searching stems each word in the query, counts the number of occurrences of each stem, and creates a vector. Optionally some stems are discarded (secondary term selection) if they have a high document frequency or to enforce a maximum number of stems, but we didn't discard any stems for our TREC runs this year. The index is searched for documents containing terms which stem to any of the stems of the vector.

The VECTOR_GENERATOR set option controls which stemming operations are performed by Intuitive Searching. For most Web Track runs, we used the default VECTOR_GENERATOR setting ('word!ftelp/base/single | * | word!ftelp/inflect') which assumes the English language, but for one submitted run we disabled stemming (using SET VECTOR_GENERATOR ''). (By default, SearchServer's index supports both exact matching (after some Unicode-based normalizations, such as converting to upper-case and decomposed form) and matching on stems.) For the Arabic runs, we always disabled stemming. When searching SearchServer table ARAB01A, for which Arabic morphological normalizations were applied to each word at index-time, the same normalizations were automatically applied to each query term.

Besides linguistic expansion from stemming, we did not do any other kinds of query expansion this year. For example, we did not use approximate text searching for spell-correction because the queries were known to be spelled correctly this year. We did not use row expansion or any other kind of blind feedback technique for the official runs.

4.4 Statistical Relevance Ranking

SearchServer calculates a relevance value for a row of a table with respect to a vector of stems based on several statistics. The inverse document frequency of the stem is estimated from information in the dictionary. The term frequency (number of occurrences of the stem in the row (including any term that stems to it)) is determined from the reference file. The length of the row (based on the number of indexed characters in all columns of the row, which is typically dominated by the external document), is optionally incorporated. The already-mentioned count of the stem in the vector is also used. To synthesize this information into a relevance value, SearchServer dampens the term frequency and adjusts for document length in a manner similar to Okapi [6] and dampens the inverse document frequency in a manner similar to [7]. SearchServer's relevance values are always an integer in the range 0 to 1000.

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'). SearchServer's RELEVANCE_DLEN_IMP parameter controls the importance of document length (scale of 0 to 1000) to the ranking.

4.5 Query Stop Words

Our QueryToRankings program removed words such as "find", "relevant" and "document" from the topics before presenting them to SearchServer, i.e. words which are not stop words in general but were commonly used in the topics as general instructions. For our CLEF runs this year [9], we expanded the list for several languages based on examining the CLEF 2000 topics (not this year's TREC topics). The full list for English is now as follows: "item",

"items", "find", "documents", "document", "relevant", "report", "what", "identify", "about", "discussing". (Some of these words, such as "about", were also in the mytrec.stp stopfile, so removing them was redundant.) Although they were unlikely to appear, corresponding words for other languages, e.g. the German word "dokumente", were removed if encountered. No Arabic words were in the list. This step was found to be only minor benefit for CLEF [9].

If a query returned no results, based on our experience with the TREC-9 Large Web Task last year, the reason was often that the queries consisted entirely of stop words. The most famous stopword query, "to be or not to be", is a philosophical question, so for the Web Track this year we pre-selected document WTX094-B32-167 (the Yahoo Philosophy page) to be returned if otherwise the query would return no results. (It turned out the situation came up this year for topic 531 (who and whom), which was judged to be a grammar question, and hence the Philosophy page was properly judged non-relevant.) As a more general technique, we may just return the results of the query "philosophy" for this situation in future years.

5 Results

The evaluation measures are explained in an appendix of the conference proceedings. Briefly: *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).

We use the following abbreviations for the evaluation measures in this paper:

AvgP: Average Precision
P@5, P@10, P@15, P@20, P@30: Precision after 5, 10, 15, 20 and 30 documents retrieved, respectively
Rec0, Rec30: Interpolated Precision at 0% and 30% Recall, respectively
AvgH: Average Precision just counting Highly Relevants as relevant
H@5, H@10, H@15, H@20, H@30: P@5, P@10, P@15, P@20 and P@30 just counting Highly Relevants as relevant, respectively
H0, H30: Rec0 and Rec30 just counting Highly Relevants as relevant, respectively

We refer to the scores with a fixed cutoff (P@5, P@10, P@15, P@20, P@30) as *early precision* scores. The other scores (AvgP, Rec0, Rec30), which can be influenced by results later in the result list, we call *recall-oriented* scores.

5.1 Web Track

The topic relevance task of the Web Track was to run 50 web queries against 10GB of web data and submit a list of the top-1000 ranked documents to NIST for judging.

NIST produced a "qrels" file: a list of documents judged to be highly relevant, relevant or not relevant for each topic. From these, the scores were calculated with Chris Buckley's *trec_eval* program, which counts all relevants the same, including highly relevants. To produce scores which just counted highly relevants as relevant, we ran *trec_eval* a 2nd time on a modified version of the qrels file which had the ordinary relevants filtered out, then multiplied by 50/44 (in 6 of the 50 topics, there were no highly relevants). Hence the scores focused on highly relevants are averaged over just 44 topics.

We submitted 4 runs for the topic relevance task of the Web Track: hum01t, hum01tl, hum01tlx and hum01tdlx. The run codes we used are as follows:

hum: Hummingbird
01: TREC 2001
t: title field used
d: description field used
n: narrative field used
l: linguistic expansion (stemming) enabled
x: weighting scheme squared importance of inverse document frequency and increased importance of document length normalization (i.e. RELEVANCE_METHOD 'V2:4', RELEVANCE_DLEN_IMP 750, instead of 'V2:3' and 500 respectively)

Tables 1 and 2 show various scores of our submitted Title-only runs, i.e. runs which just used the original web query. Additionally, for some measures, NIST reported the median scores of the 77 submitted Title-only runs from all groups for each of the 50 topics; we show the average of the median scores:

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
1a: hum01tlx	0.1949	38.4%	33.2%	30.53%	28.6%	25.47%	0.6168	0.2708
1b: hum01tl	0.1784	36.8%	32.2%	28.93%	26.6%	23.73%	0.5940	0.2485
1c: hum01t	0.1582	39.6%	30.8%	28.13%	26.0%	22.67%	0.5665	0.2210
Median (77 runs)	0.1402	n/a	26.6%	n/a	22.5%	20.53%	n/a	n/a

Table 1: Precision of Submitted Title-only runs counting all relevants the same

Run	**AvgH**	**H@5**	**H@10**	**H@15**	**H@20**	**H@30**	**H0**	**H30**
2a: hum01tlx	0.1909	18.6%	13.0%	11.51%	9.9%	7.81%	0.4186	0.2417
2b: hum01tl	0.1855	18.6%	13.0%	10.76%	9.3%	7.58%	0.3890	0.2235
2c: hum01t	0.1684	20.9%	14.3%	11.36%	9.5%	7.73%	0.3807	0.2116

Table 2: Precision of Submitted Title-only runs just counting Highly Relevants as relevant

Impact of Stemming (compare hum01t to hum01tl): When counting just highly relevants as relevant, the early precision scores (P@5, P@10, P@15, P@20, P@30) were higher with stemming disabled, and the recall-oriented scores (AvgP, Rec0, Rec30) were higher with stemming enabled. This result fits intuition: stemming ought to increase recall because more word variants are allowed to match, but sometimes the variants may not reflect the query as accurately, hurting precision. When counting all relevants the same, the earliest precision score (P@5) was again higher with stemming disabled; the other early precision scores were modestly higher with linguistic expansion enabled, though by a smaller margin than for the recall-oriented scores. The difference from the result for highly relevants may be from precision suffering less when the quality of the match is not required to be as high. None of these differences were statistically significant at the 5% level by the two-sided Wilcoxon signed rank test [5].

If stemming helps recall but hurts early precision, it may be better to give a higher weight to the original query word than the generated variants. We haven't yet run any experiments with this approach.

Note that all topics were English. In our CLEF 2001 experiments this year, we found that the impact of SearchServer's stemming was normally larger in other European languages, particularly in German and Dutch. [9]

There are two differences in the weighting scheme between the submitted hum01tl and hum01tlx runs. To isolate the impact of each change, Tables 3 and 4 show diagnostic runs whose settings are the same as for hum01tlx except for the document length importance setting (RELEVANCE_DLEN_IMP). Rows 3d and 4d are the same as rows 1a and 1b, respectively (the hum01tlx run). Rows 3c and 4c differ from hum01tl in just the relevance method (V2:4 vs. V2:3):

DLen Importance	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
3a: 0	0.1289	21.6%	21.0%	18.93%	17.7%	16.80%	0.3896	0.1828
3b: 250	0.1927	36.0%	31.6%	29.20%	26.6%	23.80%	0.5967	0.2708
3c: 500	0.2004	39.6%	34.8%	32.13%	29.3%	25.47%	0.6178	0.2803
3d: 750	0.1949	38.4%	33.2%	30.53%	28.6%	25.47%	0.6168	0.2708
3e: 1000	0.1725	34.8%	30.8%	28.27%	26.3%	23.33%	0.5663	0.2326

Table 3: Impact of Document Length Normalization (Title-only runs)

DLen Importance	**AvgH**	**H@5**	**H@10**	**H@15**	**H@20**	**H@30**	**H0**	**H30**
4a: 0	0.1084	10.0%	8.6%	6.67%	6.5%	5.31%	0.2259	0.1594
4b: 250	0.1795	17.7%	12.7%	11.22%	9.8%	8.18%	0.3447	0.2194
4c: 500	0.2000	20.5%	14.3%	11.97%	9.8%	8.56%	0.4228	0.2516
4d: 750	0.1909	18.6%	13.0%	11.51%	9.9%	7.81%	0.4186	0.2417
4e: 1000	0.1595	14.5%	11.4%	10.15%	9.1%	7.35%	0.3210	0.2131

Table 4: Impact of Document Length Normalization (Title-only runs) on Highly Relevants

Impact of Document Length Normalization: Ignoring document length (rows 3a and 4a) hurt all scores; average precision was 30-55% higher in the other rows, and Precision@10 was 45-65% higher in the other rows. The impact on highly relevants was even larger; average precision was up to 85% higher in the other rows. For most measures, a setting of 500 produced the highest scores of the settings investigated. When comparing the document length importance setting of 500 with 0 (i.e. compare 3c to 3a, and 4c to 4a), the differences in all of the shown measures (i.e. from AvgP through H30) are statistically significant at the 1% level by the two-sided Wilcoxon signed rank test.

Impact of squaring the importance of inverse document frequency (compare 1b to 3c, and 2b to 4c, which are the same except for the relevance method (V2:3 vs V2:4)): All measures were higher with the importance of inverse document frequency squared (relevance method V2:4). The differences were statistically significant at the 1% level for AvgP, P@10 and P@20, and at the 5% level for P@15, P@30, Rec30, AvgH, H@30 and H30, by the two-sided Wilcoxon signed rank test. Note that on some other test collections, such as the CLEF news collections, we have seen V2:3 receive higher scores.

For the benefit of the relevance assessment pools, we donated one run with the Description field included (hum01tdlx) and assigned it highest judging priority. Tables 5 and 6 show scores for hum01tdlx and a diagnostic run which is the same as hum01tdlx except that the Narrative field was also included. Table 5 also shows averages of the medians reported by NIST which were based on a group including all submitted non-Title-only runs, including 2 manual runs and some runs using the Narrative.

Impact of including the Description field (compare hum01tlx to hum01tdlx, i.e. 1a to 5a, and 2a to 6a): All scores were higher when including the Description. The differences were statistically significant at the 5% level for AvgP, P@20, Rec0, H@30 and H30 by the two-sided Wilcoxon signed rank test. None of the differences were statistically significant at the 1% level.

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
5a: hum01tdlx	0.2201	44.4%	36.6%	33.73%	32.2%	28.33%	0.7207	0.2977
5b: 5a + Narr	0.2362	46.8%	39.0%	35.07%	33.3%	28.93%	0.7361	0.3252
Median (20 runs)	0.1877	n/a	37.2%	n/a	31.2%	26.87%	n/a	n/a

Table 5: Precision of non-Title-only runs counting all relevants the same

Run	**AvgH**	**H@5**	**H@10**	**H@15**	**H@20**	**H@30**	**H0**	**H30**
6a: hum01tdlx	0.2082	20.5%	15.0%	12.73%	11.4%	9.69%	0.4478	0.2913
6b: 6a + Narr	0.1950	20.9%	14.5%	13.18%	11.8%	9.63%	0.4241	0.2647

Table 6: Precision of non-Title-only runs just counting Highly Relevants as relevant

Impact of including the Narrative field (compare 5a to 5b, and 6a to 6b): When counting all relevants the same, all investigated scores were higher when including the Narrative. When just counting highly relevants as relevant, most of the scores were lower when including the Narrative. None of the differences were statistically significant at the 5% level by the two-sided Wilcoxon signed rank test.

Table 7 shows per-topic comparisons of our submitted runs with the medians in their category for the measures reported by NIST: Average Precision, Precison@10, Precision@20 and Precision@30, respectively. In each comparison, we show the number of topics on which the run scored higher than the median, lower than the median and tied with the median (Higher-Lower-Tied). Differences statistically significant at the 1% level by the two-sided Wilcoxon signed rank test are marked with two asterisks (**), and differences just significant at the 5% level are marked with a single asterisk (*):

Run	AvgP	P@10	P@20	P@30
hum01tlx	41-8-1 **	23-9-18 **	27-7-16 **	28-4-18 **
hum01tl	36-13-1 **	21-4-25 **	22-10-18 **	24-11-15 **
hum01t	28-21-1 *	21-9-20 *	23-11-16 *	25-13-12
hum01tdlx	31-17-2 **	14-15-21	18-15-17	20-15-15

Table 7: Per-topic comparison of Submitted runs with Medians

The per-topic comparisons show a lot of ties in the early precision scores, particularly P@10, because of the small number of documents considered. Still, in each measure, the difference of the hum01tlx and hum01tl runs with the medians is statistically significant at the 1% level by the two-sided Wilcoxon signed rank test (the calculation of the significance level discards the ties, following [5]).

The significance level (p-value) for the two-sided Wilcoxon signed rank test defined in [5] was computed by our own implemented algorithm. The computation is exact (aside from double-precision roundoff errors) even in the case of tied absolute differences. For the Wilcoxon signed rank test we assume that the differences on differing topics are independent, and that the differences are from a distribution which is symmetric about a median difference. The test tests the hypothesis that the median difference is zero. For more details, see [5].

5.2 Cross-Language Track

Table 8 shows our submitted Arabic runs, which were all monolingual runs, i.e. used the Arabic versions of the topics. The baseline run was humAR01td, a Title+Description run which applied some experimental Arabic

morphological normalizations to the words before indexing. The other runs were the same as the baseline except for one factor. Run humAR01tdm disabled the Arabic-specific normalizations (but not general ones such as case normalization). Run humAR01tdx used a different weighting scheme which squared the importance of inverse document frequency and also increased the adjustment for document length. Run humAR01t was Title-only. Run humAR01tdn was Title+Description+Narrative. No stop words were applied:

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
8a: humAR01td	0.2441	50.4%	49.2%	47.73%	46.2%	43.07%	0.7494	0.3149
8b: humAR01tdm	0.2087	48.0%	48.4%	48.27%	46.2%	41.20%	0.7238	0.2848
8c: humAR01tdx	0.2465	51.2%	48.0%	46.13%	43.8%	39.07%	0.7486	0.3235
8d: humAR01t	0.2663	53.6%	48.8%	48.0%	45.0%	41.33%	0.7755	0.3390
8e: humAR01tdn	0.2395	62.4%	51.6%	49.33%	45.8%	43.47%	0.8312	0.2823

Table 8: Precision of Submitted Monolingual Arabic runs

Impact of Arabic morphological normalizations (compare humAR01tdm to humAR01td): All investigated scores except P@15 were tied or higher when the Arabic morphological normalizations were applied. None of the differences were statistically significant at the 5% level by the two-sided Wilcoxon signed rank test. (The Arabic test collection just contained 25 topics, making it harder to detect significant differences than for the web collection, which had 50 topics.)

Impact of changing the weighting scheme (compare humAR01td to humAR01tdx): It made little difference.

Impact of excluding the Description field (compare humAR01td to humAR01t): Some scores were higher when just using the Title field (AvgP, P@5, P@15, Rec0, Rec30). Others were higher when the Description was included (P@10, P@20, P@30). It was noted at the conference that the Titles for these topics were a little longer than usual.

Impact of including the Narrative field (compare humAR01td to humAR01tdn): Some scores were higher when including the Narrative (P@5, P@10, P@15, P@30, Rec0) but a few were lower (AvgP, P@20, Rec30).

After the conference, we added approximately 2000 Arabic stop words based on the ISI list [8]. Table 9 shows the new scores when redoing runs humAR01td, humAR01t and humAR01tdn with the stop words (note: an experimental version of SearchServer 5.3 (pre-release) was used for the Arabic diagnostic runs, but its document ranking with respect to Arabic was the same as SearchServer 5.0's except for the experimental differences described in this section):

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
9td: 8a + stp	0.2617	50.4%	52.0%	48.27%	47.8%	43.87%	0.7558	0.3415
9t: 8d + stp	0.2692	52.0%	48.0%	45.87%	43.0%	40.27%	0.7624	0.3523
9tdn: 8e + stp	0.2639	60.8%	54.4%	51.47%	50.6%	44.40%	0.8631	0.3335

Table 9: Precision of runs using Arabic Stop Words

Impact of Arabic stop words (compare 8a to 9td, 8d to 9t, and 8e to 9tdn): While the scores just increased modestly, the increases were consistent across topics. For example, in the Title+Description case (runs 8a vs 9td), 22 topics had a higher score in average precision, just 1 lower and 2 tied, when using the stop words. In the Title+Description case, the difference in average precision was statistically significant at the 1% level, and the differences in P@10 and P@20 were statistically significant at the 5% level, by the two-sided Wilcoxon signed ranked test.

As another experiment, we added some morphological normalizations mentioned by other groups which we weren't already using, particularly the orthographic variation mentioned by BBN [11] (YEH vs. ALEF MAKSURA at end of word) and removing WAW prefixes as mentioned by Berkeley [3]. Table 10 shows the scores when redoing the runs of Table 9 with the additional rules:

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
10td: 9td +rules	0.2831	57.6%	51.2%	48.53%	45.8%	43.60%	0.7917	0.3837
10t: 9t +rules	0.2939	57.6%	49.6%	47.47%	46.6%	42.80%	0.8269	0.3853
10tdn: 9tdn+rules	0.2802	62.4%	56.8%	51.47%	49.4%	45.07%	0.8746	0.3646

Table 10: Precision of runs with Additional Arabic Morphological Normalizations

Impact of additional Arabic morphological normalizations (compare 9td to 10td, 9t to 10t, and 9tdn to 10tdn): Most of the scores modestly increased from the new rules. Focusing on the Title+Description case, the difference in P@5 was statistically significant at the 5% level by the two-sided Wilcoxon signed rank test, but the other differences were not.

Combined impact of (updated) Arabic morphological normalizations and stop words (compare 8b to 10td): The combination of Arabic morphological normalizations (including the experimental updates) and the stop words increased average precision by 36% and the difference in average precision was statistically significant at the 1% level by the two-sided Wilcoxon signed rank test. None of the other differences were statistically significant even at the 5% level. Precision@10 just increased 6%. Early precision may benefit less from normalization rules because there may be enough exact matches to find.

Some groups found that query expansion worked well on this collection, so we applied the "row expansion" technique described in last year's paper [10]. Roughly speaking, row expansion is a pseudo-relevance feedback technique in which it is assumed that the top rows of the initial query are relevant and SearchServer's Intuitive Searching uses them to generate new, broader queries. (In practice, a SearchServer user would specify which rows are relevant, which should produce better results than the "blind" automatic technique applied here.) Like last year, we used the top-5 rows; a minor difference is that for the row expansion queries, we used a document frequency parameter of 5% (i.e. RELEVANCE_METHOD 'V2:3:05') instead of the experimental secondary term selection approach used last year. Table 11 shows the scores after applying row expansion to the runs of Table 10:

Run	**AvgP**	**P@5**	**P@10**	**P@15**	**P@20**	**P@30**	**Rec0**	**Rec30**
11td: 10td +exp	0.3185	59.2%	52.8%	52.53%	50.6%	46.80%	0.7918	0.4430
11t: 10t +exp	0.3268	58.4%	56.0%	53.33%	50.6%	47.47%	0.8070	0.4262
11tdn: 10tdn+exp	0.3285	63.2%	60.4%	55.47%	53.8%	50.13%	0.8684	0.4351

Table 11: Precision of Arabic runs after Row Expansion

Impact of row expansion (compare 10td to 11td, 10t to 11t, and 10tdn to 11tdn): Most of the scores modestly increased from row expansion; average precision was up 11-17%. Focusing on the Title+Description case, the differences in average precision and Rec30 were statistically significant at the 1% level by the two-sided Wilcoxon signed rank test; the other differences were not statistically significant at even the 5% level. Query expansion techniques such as row expansion may help recall-oriented measures by contributing terms from the top documents which are not automatically generated from the initial query.

Combined impact of (updated) Arabic morphological normalizations, stop words and row expansion (compare 8b to 11td): Applying all the techniques described above increased average precision by 53%, but increased Precison@10 by just 9%. The differences in average precision and Rec30 were statistically significant at the 1% level, and the difference in Precision@5 was statistically significant at the 5% level, by the two-sided Wilcoxon signed rank test. None of the other differences were statistically significant at the 5% level.

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] Peter Bailey, Nick Craswell and David Hawking, Engineering a multi-purpose test collection for Web retrieval experiments (DRAFT, accepted, subject to revision, by Information Processing and Management). http://www.ted.cmis.csiro.au/TRECWeb/

[3] Aitao Chen and Frederic Gey. (University of California at Berkeley.) Translation Term Weighting and Combining Translation Resources in Cross-Language Retrieval. Notebook paper in draft TREC 2001 Conference Proceedings.

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

[5] Myles Hollander and Douglas A. Wolfe. *Nonparametric Statistical Methods.* Second Edition, 1999. John Wiley & Sons.

[6] 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

[7] Amit Singhal, John Choi, Donald Hindle, David Lewis and Fernando Pereira. AT&T at TREC-7. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Seventh Text REtrieval Conference (TREC-7).* NIST Special Publication 500-242. http://trec.nist.gov/pubs/trec7/t7_proceedings.html

[8] 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

[9] Stephen Tomlinson. Stemming Evaluated in 6 Languages by Hummingbird SearchServer™ at CLEF 2001. To appear in Carol Peters, editor, Proceedings of the Cross-Language Evaluation Forum (CLEF) 2001, Darmstadt, Germany, September 2001. Springer Lecture Notes for Computer Science (LNCS) series.

[10] 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

[11] Jinxi Xu, Alex Fraser and Ralph Weischedel. TREC 2001 Cross-lingual Retrieval at BBN. Notebook paper in draft TREC 2001 Conference Proceedings.