SMART High Precision : TREC 7

Chris Buckley*, Mandar Mitra†, Janet Walz*, Claire Cardie†

Abstract

The Smart information retrieval project emphasizes completely automatic approaches to the understanding and retrieval of large quantities of text. We continue our work in TREC 7, concentrating on high precision retrieval. In particular, we present an in-depth analysis of our High-Precision Track results, including offering evaluation approaches and measures for time dependent evaluation. We participated in the Query Track, making initial efforts at analyzing query variability, one of the major obstacles for improving retrieval effectiveness.

Basic Indexing and Retrieval

In the Smart system, the vector-processing model of retrieval is used to transform both the available information requests as well as the stored documents into vectors of the form:

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{it})$$

where D_i represents a document (or query) text and w_{ik} is the weight of term T_k in document D_i. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms actually assigned. The assumption is that t terms in all are available for the representation of the information.

The basic "tf*idf" weighting schemes used within SMART have been discussed many times. For TREC 7 we use the same basic weights and document length normalization as were developed at Cornell by Amit Singhal for TREC 4([3, 5]. Tests on various collections show that this indexing is reasonably collection independent and thus should be valid across a wide range of new collections. No human expertise in the subject matter is required for either the initial collection creation, or the actual query formulation.

The same phrase strategy (and phrases) used in all previous TRECs (for example [2, 3, 4, 1]) are used for TREC 7. Any pair of adjacent non-stopwords is regarded as a potential phrase. The final list of phrases is composed of those pairs of words occurring in 25 or more documents of the initial TREC 1 document set. Phrases are weighted with the same scheme as single terms.

When the text of document D_i is represented by a vector of the form $(d_{i1}, d_{i2}, \ldots, d_{it})$ and query Q_j by the vector $(q_{j1}, q_{j2}, \ldots, q_{jt})$, a similarity ($S$) computation between the two items can conveniently be obtained as the inner product between corresponding weighted term vectors as follows:

$$S(D_i, Q_j) = \sum_{k=1}^{t} (d_{ik} * q_{jk}) \tag{1}$$

Thus, the similarity between two texts (whether query or document) depends on the weights of coinciding terms in the two vectors.

The Cornell TREC experiments use the SMART Information Retrieval System, Version 13.2, and most were run on a dedicated Intel dual 200 Mhz Pentium Pro running Solaris, with 512 Megabytes of memory

*SabIR Research

†Department of Computer Science, Cornell University, Ithaca, NY 14853-7501

and 49 Gigabytes of local disk (some runs were made on a Sun UltraSparc 1/140 with 512 Megabytes of memory).

SMART Version 13 is the latest in a long line of experimental information retrieval systems, dating back over 30 years, developed under the guidance of G. Salton. The new version is approximately 44,000 lines of C code and documentation.

SMART is highly flexible and very fast, thus providing an ideal platform for information retrieval experimentation. Documents for TREC 7 are indexed at a rate of about 2 Gigabytes an hour, on hardware costing under $10,000 new. Retrieval speed is similarly fast, with basic simple searches taking much less than a second a query.

High-Precision Track

Track Overview

TREC 7 is the second year the High-Precision (HP) track been run. It is an attempt to perform a task that is much more closely related to real-world user interactions than the ad-hoc or routing task. The goal is simple: a user is asked to find 15 relevant documents in 5 minutes. No other restrictions are put on the user (other than no prior knowledge of the query, and no asking other users for help). Official evaluation is simply how many actual relevant documents were found among the 15 documents supplied by the user, modified slightly for those queries with fewer than 15 relevant documents in the collection (Relative Precision at 15 documents).

There are no restrictions on the type of resources the user may use during this task other than

- Only one user per query per run (no human collaboration).
- The user and system can have no previous information about the query (eg, the system cannot have previously built a query dependent data structure.)

In particular, the users are allowed to make multiple retrieval runs, allowed to look at documents, allowed to use whatever visualization tools the system has, and allowed to use system or collection-dependent thesauruses, as long as they stay within the 5 minute clock time.

This track tests (at least) the effectiveness, efficiency, and user interface of the systems. The task provides a forum for testing many of the neat ideas in user interface and visualization that have been suggested over the years.

Unlike other interactive evaluations (for example, the TREC 6 Interactive task), no attempt is made to factor out user differences when comparing across systems. All users are assumed to be experts and equally proficient in use of their own system. This allows for fair comparison of systems, but implies that the absolute level of performance within the track will be better than the level obtainable from casual users. These are upper-bound interactive experiments.

The only changes in the rules from the TREC 6 track are to raise the number of relevant documents required to 15 instead of 10, and to forbid cutting and pasting of the original query. This latter change requires the participants to type in the query, and makes the task fairer for those groups for whom cutting and pasting would not give a query in the proper form. It also has the side effect of making the task more difficult since reading and typing the query might take 30 seconds (10% of the available time).

High-Precision Methodology

Our methodology for the TREC 7 HP task is very similar to those we've used in the past 3 TRECs. [3, 4, 1]. The user's main task is to provide relevance judgements to be fed to our standard Rocchio relevance feedback algorithm. Direct modification of the query (adding/deleting terms to/from the query or directly modifying weights) was also occasionally (rarely) used by the searchers. The other principal component of our technique is the use of pipelining or "parallel" processing so that expensive retrieval techniques can be executing while the user continues to make judgements. The details of the method are given below:

1. The current time is noted. The user views the topic supplied by NIST and types a query into the system.
2. The query entered by the user is indexed and a set of documents is retrieved using a simple vector match.
3. The top-ranked documents are presented to the user.
4. The user starts viewing the documents and judging them 'relevant', 'non-relevant' or 'possibly relevant'.

   In parallel, a child process is forked to retrieve additional documents using a more sophisticated retrieval algorithm: the initial query is used to retrieve 1000 documents, the top 20 are assumed to be relevant, documents ranked 501-1000 are assumed to be non-relevant, and automatic feedback is used to expand the query by 25 single terms and 5 phrases, using $\alpha = 8, \beta = 8$ and $\gamma = 8$.
5. After every judgement, the current time is noted. All documents retrieved so far are sorted such that the documents judged relevant come first, followed by all documents judged possibly relevant, followed by all unjudged documents, and the top 15 documents in this ranking are saved in a file and time-stamped.
6. After every 5 categorical judgements (i.e. 'relevant' or 'non-relevant'), a relevance feedback process is started in parallel if the child process is idle. For this process, documents marked relevant by the searcher are assumed to be relevant, and documents marked non-relevant as well as those retrieved at ranks 501-1000 by the initial user query are assumed to be non-relevant. Documents marked "possibly relevant" are not used in the feedback process. The query is expanded by 25 words and 5 phrases. $\alpha = 8, \beta = 8$ and $\gamma = 8$ are used. While this feedback process is running in the background, the user continues to judge more documents.
7. When the child process is done (i.e. retrieval or feedback completes), and the new retrieval results are available, these results are merged into the current list of top-ranked documents being shown to the user.
8. The final top 15 documents for the query will be the last set of 15 documents saved with a timestamp under the 5-minute limit

User Interface. The user interface for the TREC 7 high-precision runs is a simple GUI using Tk/Tcl. The display has 4 main windows

1. Text of user query
2. Vector form of user query
3. Current titles being judged
4. Current document, with query terms optionally highlighted

The GUI is used to view documents and mark documents 'relevant', 'non-relevant', or 'possibly relevant'. As soon as one document is judged, the next document is displayed. The user can go back and re-judge previously judged documents if needed, though in practice this was done mostly to correct errors of clicking the wrong judgement button. The interface may also be used to modify the user query statement by either modifying the text, or by modifying the term weights. After modification, the new query (or query vector) is used as the user query and combined with the existing relevance judgements in a relevance feedback retrieval. As an aid to pacing the query session, the interface displays the time elapsed since the beginning of the search.

Users and Settings. Three runs are presented; each the result of one user running all 50 queries. The users and some environmental characteristics are:

1. User 1 - Run HP1
   - Experience: System designer (HP interface designer)
   - Machine: Pentium Pro 200 dual processor with Solaris
   - UI Settings: Highlighting of terms on
2. User 2 - Run HP2
   - Experience: SMART System implementer
   - Machine: UltraSparc 1/140
   - UI Settings: Highlighting of terms on
3. User 3 - Run HP3
   - Experience: System designer
   - Machine: UltraSparc 1/140
   - UI Settings: Highlighting of terms on

All three users should be considered experts and were running on comparable machines, though the Pentium was slightly faster.

Effectiveness Results.

The effectiveness evaluation results are presented in Table 1. The base case is the official run Cor7A3rff which gives the precision at 15 documents of that automatic run. All three runs do very well and are amazingly close to each other. Less than 1% separates the top run from the bottom run. The top run is greater than or equal to the median on 86% of the queries, though the second run is best on more queries.

Run	Precision	Relative Precision	Num queries Best	Num queries $\geq$ Median
Base	.4760	-	-	-
Cor6HP1	.5787	.5909	12	38
Cor6HP2	.5813	.5920	19	37
Cor7HP3	.5853	.5967	16	43

Table 1: High-Precision comparison (50 queries)

Agreements with TREC Assessors.

One important question is how the users agree with the official TREC relevance judgements. If the HP track is to have meaning, the disagreement between user interpretation of relevance to a query, and the official assessor interpretation can not dominate the results. Table 2 gives the total number judged relevant, possibly relevant, and non-relevant for each user, for both the TREC-assessor judged relevant documents and the TREC-assessor judged non-relevant documents. For example, User 3 judged 290 documents relevant (159) or iffy (131) that the official assessors had judged non-relevant.

Run	TREC judged Rel			TREC judged NonRel			Overlap
	UserRel	Iffy	NonRel	UserRel	Iffy	NonRel	(Iffy=rel)
Cor7HP1	315	170	51	79	181	448	61%
Cor7HP2	396	73	36	115	128	444	63%
Cor7HP3	374	100	84	159	131	674	56%

Table 2: High-Precision User-assessor consistency (50 queries)

The last column gives the overlap on judgements of relevant documents. If "Iffy" documents are assumed to be relevant, then the overlap for User 1 is 61% (from (315+170) / (315+170+51+79+181)). This is noticeably less than in previous studies, though it is not clear how much of this is due to the task. Often users marked documents as "Iffy" just because they were the only documents seen that were close to being relevant. Note that if we define "Iffy" documents as non-relevant when calculating overlap, the values are even lower: User 3 would have an overlap of only 52%.

The great majority of the disagreements are the users considering documents relevant that the assessor considered non-relevant. In fact, consider the 15 queries with lowest overlap for each of the three users; for all 45 queries the user has looser criteria than the assessor. This is to be expected, since the assessor as the originator of the query can easily have in mind a stricter query than made it to the topic description. For example, in query 375 "hydrogen energy", the assessor obviously did not want hydrogen fuel for car engines, though that wasn't clear from the topic. The three users marked a total 50 documents as relevant or iffy that were not relevant. Query 363 "tunnel disasters" was another with major disagreements (36 documents).

The disagreements in the other direction are rarer and a bit less obvious. For example, query 377, "cigar smoking", had the most disagreements, with 15 total assessor relevant documents being marked non-relevant by the three users.

The overall level of disagreement between assessor and users is unfortunately high. The overall level of performance is being strongly affected by agreement with assessor, rather than intrinsic performance.

Difficulty of Task.

One of the ways of telling how easy or difficult the TREC 7 HP task is, is to look at the queries for which the users did not find 15 documents that they thought were relevant. Table 3 gives the number of documents that are included in the final submitted retrieval without being judged. There will be unjudged documents only if the user did not find 15 relevant or iffy documents after 5 minutes.

According to the logs, it is obvious the users simply ran out of time on several queries. For example, for query 397, User 3 had just focused in on a set of relevant documents. User 3 had found 8 relevant or iffy documents in 5 minutes, so 7 documents were filled in. 6 out of those 7 were relevant. Similarly, for Query 377, User 2 had only found 6 relevant or iffy documents by the end of 5 minutes, but 4 out of the next 9 documents were relevant. For these few queries, it is clear the 5 minute limit was effective and stressed the system. These queries account for the comparatively high number of relevant documents among the unjudged (ranging from 10% to 16%).

Run	num docs unjudged	num queries with unjudged	num unjudged rel docs
Cor7HP1	122	24	12
Cor7HP2	139	25	20
Cor7HP3	84	17	13

Table 3: Unjudged Retrieved Documents

However, half or less of the 50 queries have any unjudged documents at all for all three users. This includes queries for which there were fewer than 15 relevant documents in the collection. This implies for the majority of the queries, the only evaluation differences are due to disagreement with assessors rather than effectiveness of system. Combined with the high disagreement between users and assessors, the conclusion must be reached that the task is too easy.

Query Analysis.

Table 4 gives some facts and timings for query construction and retrieval runs. User 2 constructed shorter initial queries and used 10 seconds less time doing so. After initial queries were constructed, the initial simple run took less than 1 second to run (timings for these runs were measured in seconds so we do not have exact figures for the initial run). While the user was perusing the initial returned documents, a complex run was taking between 11 and 16 seconds. Then there were an average of 5 feedback runs made per query, each one taking from 7 to 12 seconds.

Unlike our TREC 6 experiments, the complex and feedback runs took a reasonably short time to complete. The user typically only had time to judge one or two documents during these runs before the new documents

Run	num query terms	Construct query time	Complex run time	Num runs Feedback	Feedback run time	Num Judged
Cor7HP1	5.34	49.2	11.1	5.06	7.6	24.9
Cor7HP2	3.44	39.7	14.1	4.7	12.2	23.8
Cor7HP3	6.82	50.5	16.2	5.3	12.4	30.4

Table 4: Query Timing and Stats

would become available. It would have been possible to have had many more feedback runs; perhaps next time we will do so.

As can be inferred from Table 4 and Table 2, User 3 took an approach of judging as many documents as possible, as fast as possible. If the document wasn't obviously relevant on the first page, it was generally judged non-relevant, with the idea that there would be other more obviously relevant documents later. This allowed User 3 to judge an extra 5 to 6 documents per query as compared to the other two users. However, User 3 also had the lowest overlap with assessors, undoubtedly due to hasty judgements. User 3 looked at more relevant documents, but the inaccuracies in judgement meant the overall results remained the same as the other two users.

Timing Evaluation.

As has been indicated above, we kept track of not only what each user document judgement was, but when it occurred. Thus we can analyze the time performance of each user, and hopefully develop time-based evaluation measures that reflect the power and efficiency of systems.

The most obvious fact to look at is when the relevant documents were retrieved. Figure 1 gives the number of relevant documents retrieved during each 5 second timeslice for User 1, on average for 50 queries. The number of retrieved relevant starts off at 0 for the first 20 to 50 seconds as the user reads and types in the query. Then it steadily increases for the next minute or so and then starts slowly decreasing up until the 5 minute point is reached. There's a big hump at 300 seconds as the 15 documents to be returned get filled in with unjudged documents. In the normal course, these documents would be judged over the next few buckets.

This graph is actually evidence against the conclusion reached earlier that the task was too easy. The rate at which relevant documents are being added close to 300 seconds is still substantial. The previous evidence indicates it can't go on for much longer, and that less than half of the queries are still active. However, there is no sudden drop-off as there would be if this particular run finds too many relevant documents.

Figure 2 compares all three users on a typical single query, Query 366. The measure being plotted is precision at 15 documents. As was discussed earlier, User 2 typed in shorter queries so started judging documents earlier than the others. User 2 maintains a lead up until 180 seconds, when User 1 takes over. Then at 240 seconds, User 3 takes the lead for the last minute.

For this particular query, it is clear that User 3 has the best end result (precision after 5 minutes). But it is also clear that User 2 and possibly User 1 have better sessions: they find relevant documents sooner during the first 4 minutes.

Figure 3 gives the same comparison except on the average of all 50 queries. Once again, User 2 has the lead for most of the session up until the very end when User 3 takes over. For most of the session, User 2 is about 10 seconds ahead of User 3 and 20 seconds ahead of User 1. Again, User 3 has the best end result, but User 2 had the best session.

Other evaluation measures give the same overall results. For example, Unranked Average Precision at 15 documents is given in Figure 4. The curve is almost identical.

One different evaluation measure is Utility(1,-1,0,0) in Figure 5. This measures increases by 1 when a relevant document is retrieved and decreases by 1 when a non-relevant document is retrieved. It is a poor evaluation measure for the HP task. It is dominated by the retrieved non-relevant documents; i.e., those documents for which user and assessor disagree on relevance. None-the-less, the results are informative.

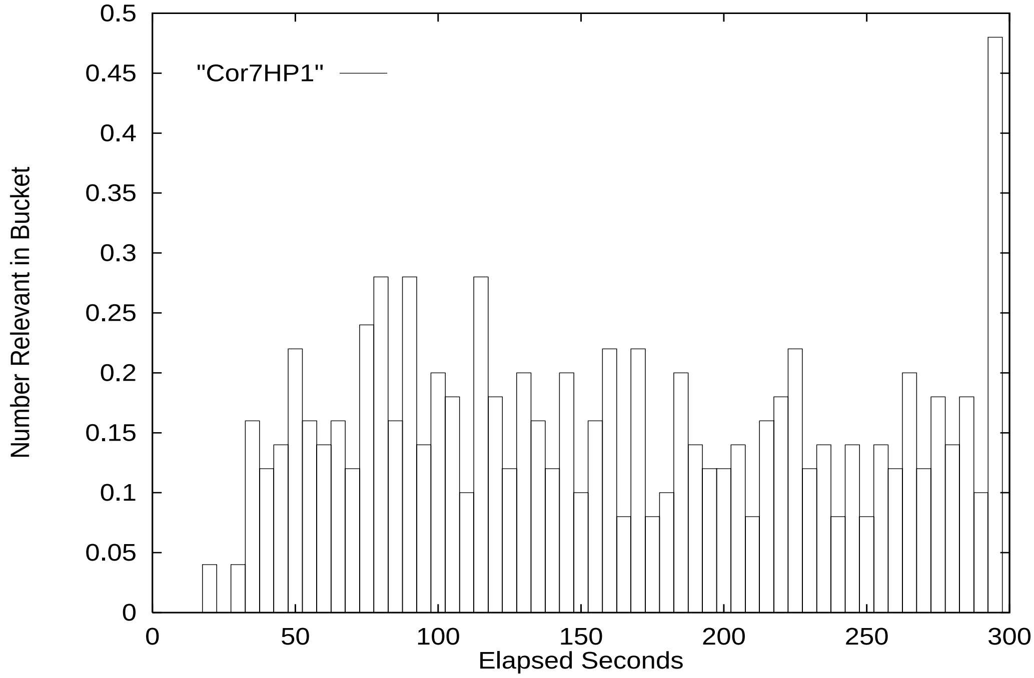

Figure 1: Average Relevant Retrieved per 5 second Timeslice over 50 Queries

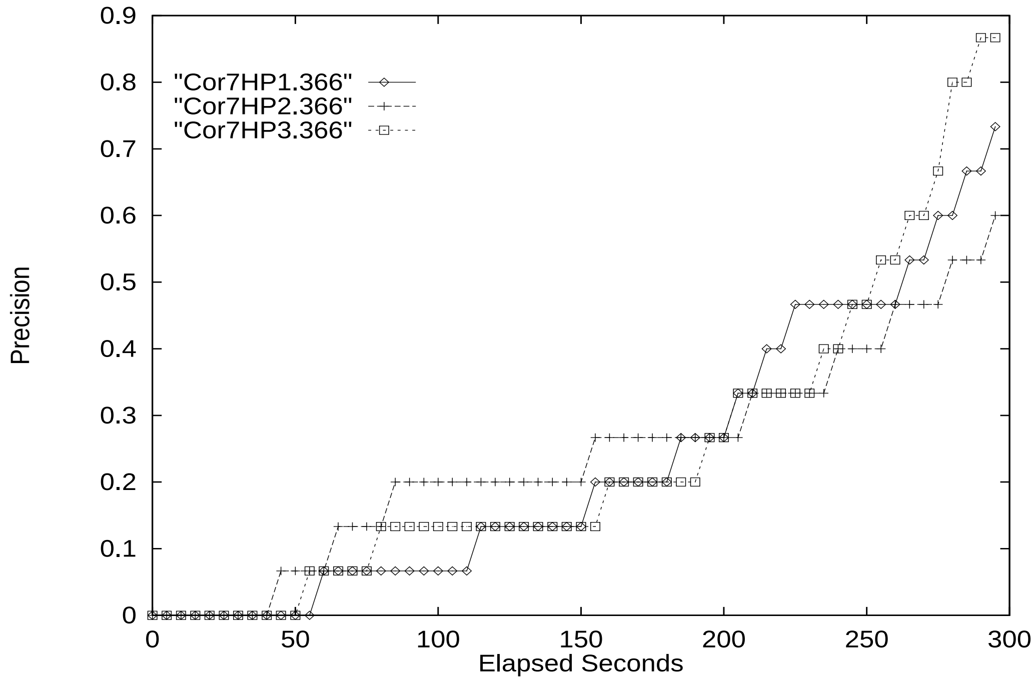

Figure 2: Precision (at 15 Documents) vs. Time for Query 366

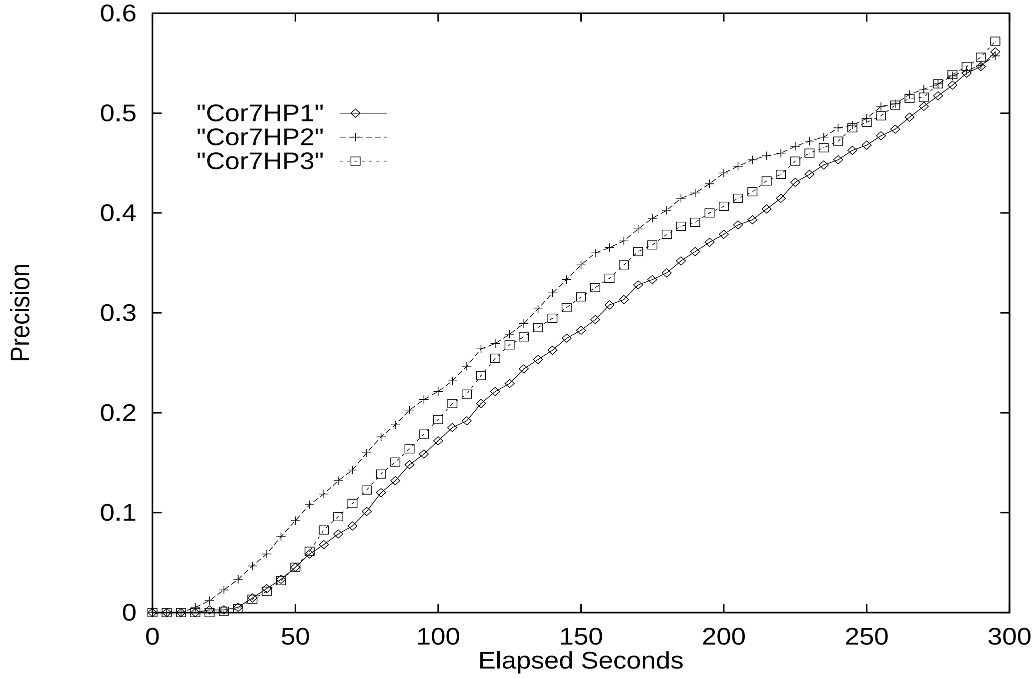

Figure 3: Precision (at 15 Documents) vs. Time over 50 Queries

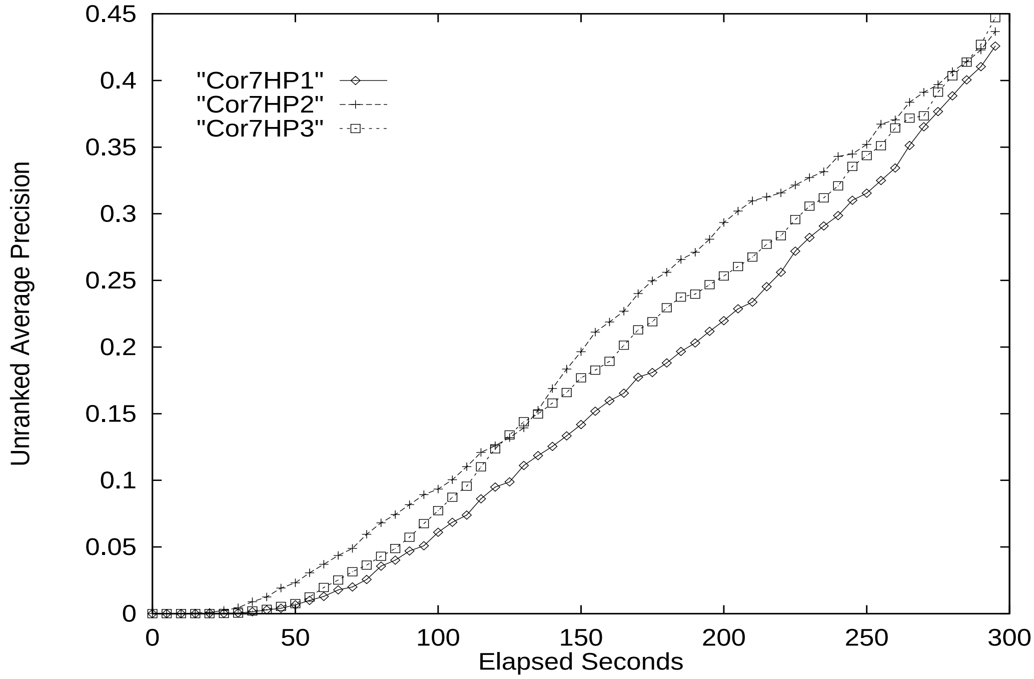

Figure 4: Unranked Average Precision vs. Time over 50 queries

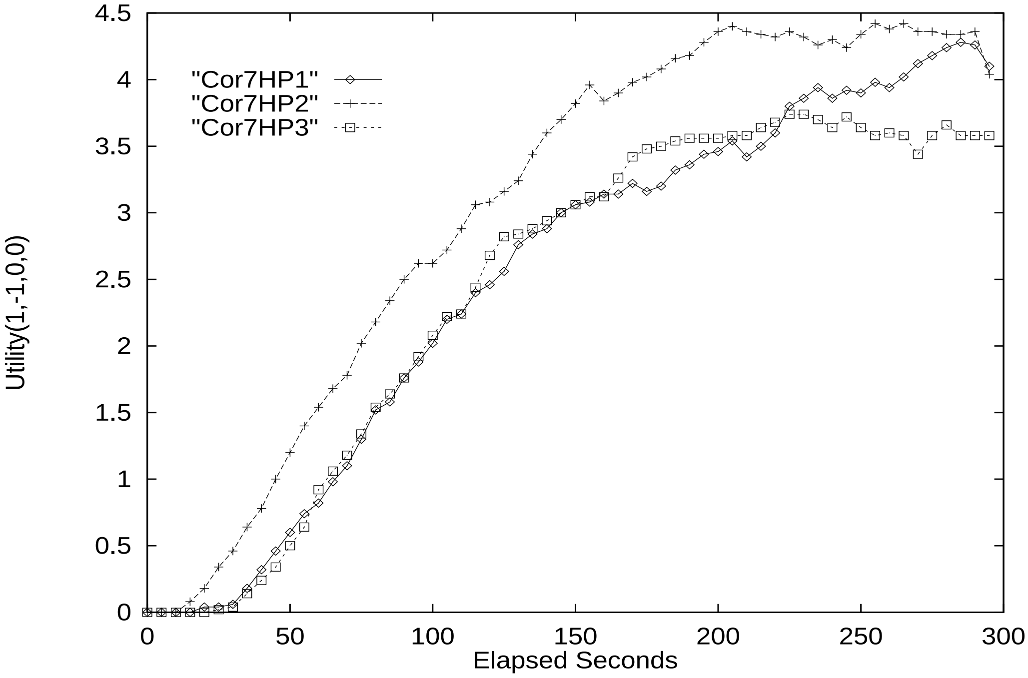

Figure 5: Utility(1,-1,0,0) vs. Time over 50 queries

User 2's lead is even more substantial (remember User 2 has the most accurate judgements as measured by agreement with assessors). But what is very interesting is how the plots for User 2 and User 3 flatten out over the last 2 minutes. For every relevant document being added, a non-relevant document is being added. This may indicate more disagreements occur late, or maybe there is a natural stopping spot late. Further study is needed, especially since the handling of "iffy" documents may be partly responsible for the effect.

All of these time-based measures and graphs suggest that a reasonable evaluation measure for an entire session is the area under each plot, much in the same way as the area under the recall-precision curve is a good single measure (this is "average precision"). Table 5 gives three such measures, corresponding to the three different plots seen above. As expected, for all 3 session measures, User 2 has a substantial (6% – 8%) lead over User 3 and even more over User 1.

Run	Average Precis	Average UAP	Average Utility(1,-1)
Cor7HP1	.2726	.1590	3.997
Cor7HP2	.3104	.1934	4.606
Cor7HP3	.2901	.1780	4.287

Table 5: Timing Evaluation

These session evaluation measures can be extended to work on any time-based retrieval. It would be very interesting to apply these measures to the standard Manual portion of the ad-hoc task. Perhaps for TREC 8, we can request that timing figures be optionally supplied, perhaps as the iteration field, to Manual submissions. There are still open questions regarding these measures. A couple that immediately spring to mind is how sensitive they are to starting time, and to size of time-slice. However, they still seem to offer a hope at bringing efficiency into evaluation of manual systems and sessions.

Note that the latest copy of trec_eval is in pub/smart/trec_eval.7.0beta.tar.gz on ftp.cs.cornell.edu and includes all the measures discussed here plus many others, though perhaps not in their final form (for instance, the timing information is assumed to be in the "sim" field but will probably be moved.)

Examples and Failure Analysis.

After the HP results were received back from NIST, the three users were asked to write a sentence or two about each query. The following comments (paraphrased in some cases) give some insights into weaknesses of the system. There were a fair number of comments about disagreements with assessors, but those are ignored here.

Query 353: Antarctica exploration

- User 3: I misspelled query as "Antartica" and didn't notice for 2 minutes (though I noticed something was wrong and revamped weights)!

Query 354: journalist risks

- User 2: I think I tried "journalist hostage" first; I got onto a single case of a journalist kidnapped by Colombian drug lords, got a bunch of non-relevant Colombian documents, and then got a few more relevant and ran out of time (first relevant document had way too many other Colombian details for the feedback to work on)

Query 371: health insurance holistic

- User 1: tough to get both aspects
- User 2: there is way too much out there on health insurance...I apparently tried just acupuncture, but couldn't get anything with insurance, and was then trying herbal medicine at the end.
- User 3: Terms were too broad and couldn't come up with good narrow terms dealing with holistic.

Query 376: world court

- User 2: the first relevant documents, about the World Court refusing to hear Libya's case, pulled up voluminous Libyan stuff that I couldn't get past

Query 381: alternative medicine

- User 1: unexpectedly difficult to get "alternative medicine"
- User 2: I probably tried "alternative medicine" first, then apparently added "nontraditional", "acupuncture" (where my first relevant document brought up all sorts of stuff on drug treatment), ...
- User 3: Couldn't find specific examples (that were judged relevant).

Query 383: mental illness drugs

- User 2: lots of articles on treating drug abuse by the mentally ill, and for some reason I didn't seem to get through as many articles as usual.
- User 3: Extremely frustrating. Never able to find that first relevant document though there are lots out there.

Query 389: illegal technology transfer

- User 1: tough because query is high-level (concept not well-represented by keywords)
- User 2: once I got a few articles, I got stuff directly on COCOM as well as violations of it and what new rules might be adopted
- User 3: Never got any relevant documents.

Overall, the comments indicate there are two system weaknesses that we may want to address in the future. The first is that for a number of queries, it is very difficult to find any relevant documents. Instead, the user spends their time plowing through piles of very similar non-relevant documents. Perhaps the user should be offered the option of "Find different documents" after a couple of iterations of normal search. The system should use the same query but come up with documents that are different from each other and from previously examined documents.

The second observed weakness is that the system occasionally becomes too focused on one sort of relevant document, and is unable to find any other sort. The "Find different documents" option should help here also. The system should emphasize the original query, and should retrieve documents different from the relevant documents seen before.

Ad-hoc Task

Over the past year since TREC 6, we tried a number of different variations of our algorithms in order to improve performance. We looked at, or re-visited, stemming, phrasing, alternative clusterings, emphasizing titles, and emphasizing beginnings of documents. Unfortunately, none of these minor variations improved performance enough to be worth adopting. This suggests we need to go back to some of our more radical variations of the past (e.g., ITL or SuperConcepts) to improve effectiveness. We ran out of time to do that this year.

Ad-hoc Methodology

The basic approach we used for this year's TREC ad-hoc task is almost identical to our TREC 6 clustering approach. Unlike in previous years, we only used one algorithm (no experimental algorithm this year!), and ran it on different topic lengths. Our TREC 6 paper [1] gives the details and rationale for the approach. The basic algorithm is

1. Retrieve 1000 documents using the initial query (using *Lnu.ltu* weights).
2. Generate cooccurrence information about the query terms from the top 1000 documents.
3. Rerank the top 50 documents as in TREC 5 (using correlation and proximity information).
4. Assume the top 20 documents relevant, documents ranked 501–1000 non-relevant.
5. Generate clusters for the top 30 documents and save the best (most heavily weighted) terms from each cluster vector.
6. Rank the cluster vectors according to their similarity to the original query (using *bnn* weights for the clusters) and select the best 2 clusters.
7. Expand the query by 25 words and 5 phrases using Rocchio expansion with $\alpha = 8$, $\beta = 8$, and $\gamma = 8$. The expansion terms are selected from among the saved terms for both clusters and the actual number of terms selected from a cluster is proportional to its similarity to the original query.
8. Retrieve the final set of 1000 documents using the expanded query.

Ad-Hoc experiments and analysis

We submitted three runs in the ad-hoc category, all using the same algorithm. Cor7A1clt uses only the title field of the topics, Cor7A2rrd uses only the description field of the topics, and Cor7A3rrf uses the entire topic description. (Note that Cor7A1clt should really be named Cor7A1rrt for consistency's sake.)

Table 6 shows the results for the various runs across 50 queries. Unlike last year, we get a very pleasing performance improvement as we increase the amount of the query we use. As always, though, the averages hide a great deal of variation at the query-by-query level. For example, the title only run scores higher than the full topic run for 19 queries; almost half! Most of those differences are small, but the full text can on occasion help immensely. For example, for query 398, "dismantling Europe's arsenal", the title only query scored .0011 in average precision but the full text query scored .5051 (and the description only query actually scored .5523).

The absolute level of performance is considerably higher than last year. Even the title only run this year beat all of our official runs last year. Given the lack of change with the system, it is obvious that the task this year was considerably easier for us.

Run	Average precision	Total rel retrieved	R precision	Precision @100 docs
Cor7A1clt	.2329	2621	.2564	.2106
Cor7A1rrd	.2543	2894	.2782	.2338
Cor7A1rrf	.2674	3198	.2953	.2584

Table 6: Ad-Hoc results (50 queries)

Table 7 shows that our runs compare reasonably with other runs. It is hard to tell much about relative performance since all automatic ad-hoc runs enter the same comparison pool this year, unlike in previous years where they were sorted by length. Even the title only run was above the median for the majority of the queries, which is impressive since it is being compared against many runs using the full topics. There are 86 runs in the comparison pool; having 5 best queries is quite respectable. However, given our lack of improvement the past two years, I suspect we are not the best.

Run	Task pool	Best	$\geq$ median
Cor7A1clt	automatic	0	27
Cor7A1rrd	automatic	3	37
Cor7A1rrf	automatic	2	42

Table 7: Comparative automatic ad-hoc results (50 queries)

Query Track

General IR research is being held up because we don't have enough queries of various types to investigate advanced retrieval techniques that are query dependent. There's no way we can get enough relevance judgements on new queries to form a good query pool. The Query track looks at multiple query variations of past TREC topics to get a large number of query formulations.

The track guideline states four goals:

1. Start investigating the split between query formation/analysis and back-end engines. Evaluating what makes a good general query formation approach.

2. Get many variations of the same topic so we can start analyzing (including with strong NLP approaches) queries, and determining what sorts of things we want to pull out of queries.

3. Get a collection of mixed fact/content queries. For decades we've had systems (eg Pnorm) that can handle these, but haven't been able to evaluate and compare due to lack of a query collection.

4. Get a collection of reasonable very short queries, more typical of real-life ad-hoc queries.

Each group forms variations of each of the 50 topics in some subsets of the following categories (as defined in the guidelines):

1. Very short: (2-3 words) based on topic.
2. Sentence: NL (natural Language), based on topic and judgements
3. Manual Feedback: Manual NL sentence based on reading 5 or so relevant documents without reference to the topic (done by someone who doesn't have the topics memorized and who might use different vocabulary than the topic). An attempt to get a sentence which might use different vocabulary than the topic.
4. Manual structured query: based on topics and judgements. Perhaps mixed fact and content queries. Perhaps result of manual NL analysis.
5. Automatic structured query: based on topics and judgements (Note that "structure" could be just a list of words, or could be very complicated based on semantics.) Perhaps the result of automatic NL analysis.

Then all groups run everybody's queries for some subset of the categories above (whatever categories their system can be made to support). The names of the submitted runs consist of 7-8 letters/digits. The first 3 letters identify the group running the query. The last 4-5 letters are the queryset id, including category. Thus, "CorAPL5a" would be Cornell running the first Category 5 query set that was constructed by APL.

Query Track Methodology

This was the first year for the query track. As it ended up, only two groups participated in the track. Thus it is impossible to come up with as many conclusions as we had wanted.

The two groups are us (Cornell/SabIR) and the APL Labs at Johns Hopkins. We constructed one set of queries in each of the 5 categories; pretty much directly using the definitions of the categories. APL constructed 4 query sets, skipping category 3 and 4, but having two versions of category 5. For the first two categories, APL deliberately tried to construct different queries than the obvious choice of words. This increased query variability, though at a cost of overall effectiveness as we will see later.

All 5 sets of queries were easy to construct. Our category 4 queries do not have much detailed structure; they are basically a weighted sum of a vector query and a pnorm query. Our category 5 queries are straight weighted relevance feedback vectors. The most difficult part of category 4 and 5 queries was reverse engineering the stemming of terms, so that we could supply weighted unstemmed terms to other groups.

The queries are all constructed in DN2 format. DN2 is a quite complicated query language, but luckily very few features needed to be known for the queries the two groups constructed. We did not run directly on the DN2 queries but translated them back and forth from normal SMART queries.

Query Track Results

Table 8 gives our results on running the 9 query set variations (5 variations from Cornell and 4 from APL). The runs all strongly differ from each other in results; depending on the evaluation measure, the differences go up to 430%. In general, the Cornell queries performed better for us than the APL queries. Part of that is that goals of the APL queries were explicitly to use different, possibly non-optimal, vocabulary. But part of it could be that we constructed queries to suit our system. In particular, the query set Cor5 was constructed using relevance feedback based on Cornell document weights. How well these weights suit other systems remains to be seen.

As normal, even with the very strong overall differences in results between query sets, large numbers of individual queries of the weaker query set do better than the corresponding query in the stronger set.

Run	Ave Prec	R Prec	NumRelRet
CorCor1	.2457	.3066	6877
CorCor2	.3367	.3901	9056
CorCor3	.2020	.2774	6690
CorCor4	.3282	.3743	8674
CorCor5	.4586	.4861	10476
CorAPL1a	.1051	.1583	4438
CorAPL2a	.1142	.1633	4239
CorAPL5a	.1971	.2600	6119
CorAPL5b	.3219	.3727	8748

Table 8: Results of Cornell Runs on Different Query Sets

Table 9 gives the number of queries (out of 50) for which one query set beats another. For instance, APL5b beat Cor2 on 38 queries, despite having weaker overall evaluation averages.

>	Cor1	Cor2	Cor3	Cor4	Cor5	APL1a	APL2a	APL5a	APL5b
Cor1	0	7	32	11	2	43	39	30	18
Cor2	43	0	46	23	4	48	47	43	22
Cor3	18	4	0	5	1	38	36	26	12
Cor4	39	27	45	0	8	48	47	41	23
Cor5	48	46	49	42	0	50	49	48	46
APL1a	6	2	11	2	0	0	27	9	2
APL2a	11	3	14	3	1	22	0	16	3
APL5a	20	7	24	9	2	40	32	0	15
APL5b	32	28	38	27	4	48	47	35	0

Table 9: Comparative Query (row better than column for X queries)

There is a tremendous amount of query variability hidden in the comparative averages. We need to understand this variability. It is not clear that 9 query variations is enough to get a handle on variability; but at least it is a start.

Comparison with past TREC's

It is difficult to determine how much systems are improving from TREC to TREC since the queries and the documents are changing. For example, in TREC 3 the "Concept" field of the queries was removed. These terms proved to be very good terms for retrieval effectiveness in TREC 1 and TREC 2; thus the TREC 3 task without them is a harder task than previous TRECs. The TREC 4 task was more difficult since so much more of the text was removed from the queries. TREC 5, TREC 6, and TREC 7 continued using short queries which seem more difficult. Also, the average number of relevant documents per query has been steadily reduced every year, going from 328 in TREC 1 to 92 in TREC 6 and 93 in TREC 7. Very broad (and easy) queries have been eliminated.

To examine both how much SMART has improved over the years of TREC, and how much harder the TREC ad-hoc tasks have gotten, we ran our 7 TREC SMART systems against each of the 7 TREC ad-hoc tasks. Actually, we present two versions of the TREC 6 task. In the first version, we use the title and description fields; in the second version we use only the description field, as we do for TREC 7. The TREC 6 description fields were unfortunately not independent of the title field.

Table 10 gives the results. Note that the indexing of the collections has changed slightly over the years so results may not be exactly what got reported in previous years. In the interest of speed, we ran our current implementation of the query and document indexing and weighting.

Comparing the columns of Table 10 gives an indication of how much harder the TREC task has gotten during the 7 years of TREC. Five quite different versions of the same system all do from 45% to 65% worse, in absolute numbers, on the TREC 7 task as compared to the TREC 1 task. The TREC 1 and TREC 2

Methodology and Run	TREC 1 Task	TREC 2 Task	TREC 3 Task	TREC 4 Task	TREC 5 Task	TREC 6 TI+DESC	TREC 6 DESC	TREC 7 DESC
TREC 1: ntc.ntc	.2442	.2615	.2099	.1533	.1048	.1224	.0997	.1137
TREC 2: lnc.ltc	.3056	.3344	.2828	.1762	.1111	.1472	.1125	.1258
TREC 3: lnc.ltc-Exp	.3400	.3512	.3219	.2124	.1287	.1499	.1242	.1679
TREC 4: Lnu.ltu-Exp	.3628	.3718	.3812	.2773	.1842	.2006	.1807	.2262
TREC 5: Exp-rerank	.3759	.3832	.3992	.3127	.2046	.2278	.1844	.2547
TREC 6: Rrk-clust	.3765	.3835	.4011	.3073	.1978	.2356	.1768	.2510
TREC 7: Rrk-clust	.3778	.3839	.4003	.3142	.2116	—-	.1804	.2543
% Change from ntc.ntc	+55	+47	+91	+105	+102	+92	+89	+124

Table 10: Comparisons of past SMART approaches with present

figures are about the same. Performance starts to drop in TREC 3 and 4 when the queries get progressively shorter. The short high-level queries of the last 3 TRECs prove very difficult for all versions of SMART.

Comparing the rows of Table 10, it is obvious that our results with our TREC 7 approach are not noticeably different from our TREC 5 or TREC 6 approach.

Conclusion

This year, Cornell and SabIR Research participated in the High-Precision and Query tracks, as well as doing the base ad-hoc task. Once again we did very well in all the tracks, ahead of the median in all tracks. (Though that does not mean all that much in the Query Track with 2 participants.)

In the High-Precision area, we looked in-depth at methods of analyzing and evaluating time-dependent retrieval sessions. We came up with several new evaluation measures that seem to capture the essentials of what a session evaluation of manual retrieval should capture. These approaches may be quite useful outside of the High-Precision track, perhaps to evaluate timed Manual retrieval.

References

[1] Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie. Using clustering and superconcepts within SMART : TREC 6. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication 500-240, 1998.

[2] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[3] Chris Buckley, Amit Singhal, and Mandar Mitra. New retrieval approaches using SMART : TREC 4. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*. NIST Special Publication 500-236, 1996.

[4] Chris Buckley, Amit Singhal, and Mandar Mitra. Using query zoning and correlation within SMART : TREC 5. In D. K. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238, 1997.

[5] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In Hans-Peter Frei, Donna Harman, Peter Schauble, and Ross Wilkinson, editors, *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29. Association for Computing Machinery, 1996.