System Summary and Timing
Organization Name: Department of Industrial Engineering,
University of Toronto
List of Run ID's: stpat1

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Methods Used to build Data Structures
    - Length (in words) of the stopword list: 24 words
    - Stemming Algorithm: No
      - Morphological Analysis: No
  - Statistics on Data Structures built from TREC Text
    - Inverted index
      - Run ID : stpat1
      - Total Storage (in MB): about 370 MB (250MB TREC text plus 120MB of index)
      - Total Computer Time to Build (in hours): about 10 hours (elapsed time)
      - Automatic Process? (If not, number of manual hours): Yes
      - Use of Term Positions? : No
      - Only Single Terms Used? : No
- Query construction
  - Manually Constructed Queries (Routing)
    - Topic Fields Used: title, desc, smry, con
    - Average Time to Build Query (in Minutes): 30 minutes
    - Type of Query Builder
      - Domain Expert: no
      - Computer System Expert: Yes (perhaps). the queries were constructed by the two researchers for this project. In comparison with other TREC-3 researchers we might not be consider computer system experts.
    - Tools used to Build Query
      - Other Lexical Tools? : Webster's online dictionary (hardly ever used in practice), Wordnet (available but not used)
    - Data Used for Building Query from
      - Training Topics: No. We didn't use a separate set of training topics. We simply used the test topics on the training database for training.
      - All Training Documents: No. Only WSJ text were used.
      - Only documents with Relevance Judgments: Yes. We used these to construct measures of precision and recall that were shown to the user each time the query was modified. The user was also able to review the list of relevant documents that were identified by the experts in the training database.
    - Method used in Query Construction
      - Term Weighting? : No.
      - Boolean Connectors (AND, OR, NOT)? : Yes
      - Proximity Operators? : Yes

· Addition of Terms not Included in Topic? : Yes

I. Source of Terms: topic definitions, documents in training database, coordinate terms, user memory

– Interactive Queries

* Initial Query Built Automatically or Manually: Manually
* Type of Person doing Interaction
  · Domain Expert: No
  · System Expert: Yes
* Average Time to do Complete Interaction
  · CPU Time (Total CPU Seconds for all Iterations): considerably less than 30 minutes, possibly between 1 and 5 minutes, but the exact value is not available. We didn't distinguish between CPU time and clock time in making our logs.
  · Clock Time from Initial Construction of Query to Completion of Final Query (in minutes): 32.44 minutes (on average)
* Average Number of Iterations: 8.9. We found it hard to define precisely what was an iteration, since this depends to a large part on the user's intention. In our system, the query was executed every time it was modified. However, we tried to assess an iteration as a "new" query rather than a simple adjustment which was one step in a sequence of adjustments to build a new iteration of the query.
* Average Number of Documents Examined per Iteration: 18.18
* Minimum Number of Iterations: 3
* Maximum Number of Iterations: 18
* What Determines the End of an Iteration: User evaluation of the effectiveness of the query followed by a change in query. Thus the definition of what an iteration is, is based entirely on the user, not the system.
  · Manual Methods

  I. Using Individual Judgment (No Set Algorithm)? : Yes. In practice there was a tendency to construct a query that had high recall and then add terms to improve precision. If this didn't work the strategy was to identify subqueries with high precision and then OR them together to create a final query that had good precision and recall.

  II. Following a Given Algorithm (Brief Description)? : No

• Searching

– Search Times

* Run ID : stpat1
* Computer Time to Search (Average per Query, in CPU seconds): not available.
  · The CPU search time was of less interest to us than the elapsed time. The search time was determined by the text retrieval engine that we were using as an off-the-shelf component of our system. We believe that we could fairly easily substitute a different retrieval engine with approximately the same features and get similar results for our interactive experiments. The whole philosophy of our approach is that

performance of this style of interactive search is largely driven by the properties and features of the user interface.

  * Component Times : not applicable.

- Machine Searching Methods
  * Boolean Matching? : Yes
  * Free Text Scanning? : Yes
- Factors in Ranking
  * Term Frequency? : Yes
  * Document Length? : Yes
  * Percentage of Query Terms which match? : Yes
- Machine Information
  * Was the Machine Dedicated or Shared: Dedicated
  * Amount of Hard Disk Storage (in MB): approximately half a gigabyte used on a 2 gigabyte disk
  * Amount of RAM (in MB): 32 MB
  * Clock Rate of CPU (in MHz): 50
- System Comparisons
  * Amount of "Software Engineering" which went into the Development of the System: Half a person year of non-professional student programming. No formal software engineering process.
  * Given appropriate resources
    - Could your system run faster? : Yes
    - By how much (estimate)? :

      I. At least twice as fast. Even if the system ran twice as fast it probably wouldn't make much difference, as thinking time, and query formulation time are the major issues. Thus getting better functionality and simplifying the user's tasks are probably better areas to look at for increasing overall system efficiency. No effort has been put into the optimization problem. Instead we attempted to identify the most useful functionality for the user based on an iterative design approach. We never stayed with one iteration long enough to justify a major optimization effort. One feature of our system is that it was originally developed as a browsing system. The information retrieval tools were specifically added to meet the needs of TREC-3. These changes were made in a hurry and not fully tested prior to the TREC-3 evaluation, thus we expect that considerable improvements can be made to the system which should lead to greatly improved retrieval effectiveness.
  * Features the System is Missing that would be beneficial: There are a number of traditional IR functions that would enhance the performance of the system greatly. These include truncation, a NOT operator, and more sophisticate relevance ranking methods. enditemize
  * Significant Areas of System
    - Brief Description of features in your system which you feel impact the system and are not answered by above questions: We think that

our system was at a general disadvantage in this evaluation because much of the query formulation that our system would be expected to excel in appeared to already have been done in the statement of the topic. For instance, the concept fields contained a number of highly diagnostic terms in several cases (e.g., chemotherapy and leukemia for topic 122). On the other hand, our system was not designed to provide relevance rankings. Thus the method that we constructed for expanded the retrieval set to 1000 documents to meet the TREC-3 requirements was extremely ad hoc. Discussions with Chris Buckley and others has convinced us that sophisticated relevance feedback algorithms will tend to be dominant in determining the overall effectiveness of a system for ranked data sets. Thus our strategy was to go with the simplest brute force method we could think of so that our results could not be attributed to the performance of a particular relevance ranking algorithm or method. One of our goals in entering the TREC-3 system was to show how a browsing style of system could be used in text retrieval. We think that systems like ours would actually look a lot better if evaluations like TREC-3 were modified to consider the specific properties of interactive retrieval. We expect that our system would do a lot better in cases where: the search topic was not precisely stated or known ahead of time, the terminology and vocabulary tended to be inconsistent or imprecise, and where the emphasis was on getting a few good articles that answer specific questions rather than as large a fraction as possible of the set of potentially relevant documents.