System Summary and Timing
Organization Name: Paracel
List of Run ID's: FDF1 FDF2

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Statistics on Data Structures built from TREC Text
    - Other Data Structures built from TREC text
      - Run ID : FDF1 FDF2
      - Type of Structure: word and phrase statistics table
      - Total Storage (in MB): 100 MB (uncompressed)
      - Total Computer Time to Build (in hours): (unrecorded)
      - Automatic Process? (If not, number of manual hours): yes
      - Brief Description of Method:
        I. Collection of in-document frequency, frequency and variance statistics for all corpus words and for all relevant document phrases. This process is only applied to the training text. For queries built one-at-a-time this process need take no additional storage.

- Query construction
  - Automatically Built Queries (Routing)
    - Average Computer Time to Build Query (in cpu seconds): 200
    - Method used in Query Construction
      - Terms Selected From
        I. All Training Documents: yes
        II. Only Documents with Relevance Judgments: yes
      - Term Weighting with Weights Based on terms in
        I. All Training Documents: yes
        II. Only Documents with Relevance Judgments: yes
      - Phrase Extraction from
        I. Only Documents with Relevance Judgments: yes

- Searching
  - Search Times
    - Run ID : FDF1
    - Computer Time to Search (Average per Query, in CPU seconds): 1462.5 (host CPU) 4547.9 (real)
    - Component Times : As above. (The search engine takes queries and text and spits out scored hits. Collection and sorting time is negligible.)
    - Run ID : FDF2
    - Computer Time to Search (Average per Query, in CPU seconds): 1068.7 (host CPU) 3097.9 (real)
    - Component Times : As above. (The search engine takes queries and text and spits out scored hits. Collection and sorting time is negligible.)
  - Machine Searching Methods

    * Probabilistic Model? : yes
    * Free Text Scanning? : yes, FDF-3 search engine
- Factors in Ranking
    * Term Frequency? : yes
    * Inverse Document Frequency? : yes
    * Other: term frequency variance, term recall, estimated query method performance
- Machine Information
    * Machine Type for TREC Experiment: Host: at various times, Sun SPARC-2, SPARC ELC, HP 9000/715 Peripheral: 4 * FDF-3/15 w/ Seagate Elite-3 and Elite-9 disks
    * Was the Machine Dedicated or Shared: shared
    * Amount of Hard Disk Storage (in MB): 3 GB (actual max usage, uncompressed data from all CD-ROMs) 22 GB (max in configuration)
    * Amount of RAM (in MB): 16 MB (32 MB swap)
    * Clock Rate of CPU (in MHz): Hosts: various FDF-3: 12.5 MHz
- System Comparisons
    * Amount of "Software Engineering" which went into the Development of the System: 0.25 person-year (on top of the underlying FDF-3 system)
    * Given appropriate resources
        · Could your system run faster? : yes
    * Features the System is Missing that would be beneficial: This is very difficult to answer. The system is currently under development. We believe that the underlying search engine has all the necessary facilities to perform the desired retrieval functions; we're currently working to make those facilities available to the user in appropriate form.
- Significant Areas of System
    * Brief Description of features in your system which you feel impact the system and are not answered by above questions:

      (a) ability to quickly re-run statistics collection.
      (b) easy reconfiguration of workbench.
      (c) automatic query selection (query-level data fusion) which also provides low-level performance data (and hence detailed performance comparisons).

      General Comments:

      There isn't place to put this in the form: we used a stop-word list for term suppression (173 words). This isn't used for searching, however, just for query construction. Also, we did statistical phrase discovery – but again, this was for query construction, not searching. CPU time is nearly meaningless. REAL TIME! Should include CPU type (which is listed elsewhere), types of disks involved, etc. Our "FDF1" run used ¿4500 sec REAL time, but ¡1500 sec CPU time, so REAL¿3*CPU! This is crucial: the user wants to know how long it took to get the answer, not how long the CPU spent

on the problem.011