System Summary and Timing
Organization Name: Bellcore
List of Run ID's: lsir1, lsir2 (routing), laia0mw, lsia0mw20f (adhoc)

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Methods Used to build Data Structures
    * Length (in words) of the stopword list: yes, 571, SMART's list
    * Stemming Algorithm:
      · Morphological Analysis: yes, SMART v11.0, modified lovins
    * Term Weighting: yes, SMART's "ltc" for both documents and queries for both adhoc and routing runs
    * Other Techniques for building Data Structures: yes LSI uses a k-dimensional vector space (k ¡¡ nterms) for representation and retrieval. LSI begins with a term-document matrix and calculates the best k-dimensional approximation to this matrix using singular value decomposition (SVD). The number of dimensions, k, was about 350 for TREC-3.
  - Statistics on Data Structures built from TREC Text
    * Other Data Structures built from TREC text
      · Run ID : lsir1, lsir2, laia0mw, lsia0mw20f
      · Type of Structure: k-dimensional vector representation (k ¡¡ nterms); one vector for each term and for each document; k 350 Routing, a 78746 term by 38175 doc sample was used to construct the LSI space. Adhoc, a 82968 term by 69997 doc sample was used to construct the LSI space, and the remaining 672,358 CD-12 documents were folded in.
      · Total Storage (in MB): Adhoc 549 Mb. Routing 109 Mb.
      · Total Computer Time to Build (in hours): Routing, about 24 hours (SVD with k=346).Adhoc, about 20 hours (SVD with k=199) plus 2 hours for folding in 672k docs.
      · Automatic Process? (If not, number of manual hours): yes
      · Brief Description of Method: LSI analysis in brief:
        1. Create term-document matrix, with user-selected term weighting (used SMART v11.0 for this pre-processing).
        2. Compute the best reduced k-dimensional approximation to this matrix using singular value decomposition (SVD). For TREC-3 about 350 dimensions were used in this approximation.
        3. If necessary, fold in any terms or documents not in the original matrix. Necessary for Adhoc runs, not for Routing runs.

- Query construction
  - Automatically Built Queries (Ad-Hoc)
    * Topic Fields Used: All
    * Average Computer Time to Build Query (in cpu seconds): about .1 sec / query (time to take weighted vector sum of query terms)
    * Method used in Query Construction

- Term Weighting (weights based on terms in topics)? : yes

- Automatically Built Queries (Routing)
  - Topic Fields Used: All (lsir1); None (lsir2 - uses only rel documents)
  - Average Computer Time to Build Query (in cpu seconds): .1-.2 sec/query (time to take weighted vector sum of query terms, lsir1 or relevant documents, lsir2)
  - Method used in Query Construction
    - Terms Selected From
      I. Topics: yes, lsir1
      II. Only Documents with Relevance Judgments: yes, lsir2 - used only relevant documents, ignored irrelevant documents
    - Term Weighting with Weights Based on terms in
      I. All Training Documents: yes

- Searching
  - Machine Searching Methods
    - Vector Space Model? : Yes, a k-dimensional vector space (k ¡¡ nterms), but matching does not use an inverted index. The cosine between the k-dim query vector and every k-dim document vector is computed. Adhoc - about 600 sec/query. Routing - about 0.1 sec/query.
  - Factors in Ranking
    - Factors in Ranking
      - Term Frequency? : yes, for constructing term-doc matrix before SVD
      - Inverse Document Frequency? : yes, for constructing term-doc matrix before SVD
      - Semantic Closeness? : yes, the cosine between the query vector and the document vector in LSI-space determines rank
      - Document Length? : yes, for constructing the term-doc matrix before the SVD
      - Other: NOTE: the ranking is determined entirely by the cosine between the query vector and the document vector. Factors like term frequency, inverse document frequency, etc. enter indirectly in the sense that they determine the cell entries for the term-doc matrix which we approximate using the k-dimensional SVD vectors.
  - Machine Information
    - Machine Type for TREC Experiment: Sparc 10
    - Was the Machine Dedicated or Shared: Shared
    - Amount of Hard Disk Storage (in MB): about 6 Gig
    - Amount of RAM (in MB): 128-384 MB, depending on machine
    - Clock Rate of CPU (in MHz): don't know
  - System Comparisons
    - Amount of "Software Engineering" which went into the Development of the System: Built as a research prototype - lots of flexibility at the cost of efficiency. About 1-2 person years.

- Given appropriate resources
  - Could your system run faster? : yes
  - By how much (estimate)? : A factor of 2-3 would be easy for the pre-processing and SVD (tho these are one time costs). Query matching is slow (linear in the number of documents) and this is trivial to parallelize - improvements of 100 times were obtained using a MasPar
  - Features the System is Missing that would be beneficial: Feature recognition (e.g., proper names, company names, geographic locations, dates, amounts of money would be especially useful for TREC); Phrases and other precision enhancing methods like proximity; Training in routing could use information about non-relevant documents; Perhaps handling "nots" would help.