System Summary and Timing
Organization Name: Cornell University
List of Run ID's: CrnlRR CrnlQR CrnlEA CrnlLA CrnlES CrnlVS

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Methods Used to build Data Structures
    - Length (in words) of the stopword list: 571 English, 342 Spanish
    - Stemming Algorithm: English - Modified Lovins, Spanish - ad-hoc
      - Morphological Analysis: Nothing Additional
    - Term Weighting: Cornell's "lnc" and "ltc" weights
    - Phrase Discovery? : Yes
      - Kind of Phrase: Adjacent words
      - Method Used (statistical, syntactic, other): Any pair of adjacent non-stopwords that occurred 25 times in D1
    - Proper Noun Identification Algorithm? : Exists, not used
    - Tokenizer? :
      - Patterns which are tokenized: nothing special
  - Statistics on Data Structures built from TREC Text
    - Inverted index
      - Run ID : CrnlEA, CrnlLA (D12,ad-hoc)
      - Total Storage (in MB): 863
      - Total Computer Time to Build (in hours): 3.6
      - Automatic Process? (If not, number of manual hours): yes
      - Use of Term Positions? : no
      - Only Single Terms Used? : no
    - Inverted index
      - Run ID : CrnlQR, CrnlRR (D3,routing)
      - Total Storage (in MB): 395
      - Total Computer Time to Build (in hours): 1.8
      - Automatic Process? (If not, number of manual hours): yes
      - Use of Term Positions? : no
      - Only Single Terms Used? : no
    - Inverted index
      - Run ID : CrnlVS, CrnlES (Spanish)
      - Total Storage (in MB): 84
      - Total Computer Time to Build (in hours): 0.38
      - Automatic Process? (If not, number of manual hours): yes
      - Use of Term Positions? : no
      - Only Single Terms Used? : yes
    - Special Routing Structures
      - Run ID : CrnlQR, CrnlER
      - Type of Structure: Occurrence statistics for the most frequently occurring (in learning set rel docs) 1000 terms for each routing query.
      - Total Storage (in MB): 1.5

      - Total Computer Time to Build (in hours): 1.0
      - Automatic Process? (If not, number of manual hours): yes
    * Special Routing Structures
      - Run ID : CrnlER
      - Type of Structure: For each query, history of how well 7 different feedback runs worked on it (learning on D1, testing on D2)
      - Total Storage (in MB): 0.1
      - Total Computer Time to Build (in hours): 1.1
      - Automatic Process? (If not, number of manual hours): yes

- Query construction
  - Automatically Built Queries (Ad-Hoc)
    * Topic Fields Used: title, nationality, narrative, factors, descriptions, concepts
    * Average Computer Time to Build Query (in cpu seconds): .03
    * Method used in Query Construction
      - Term Weighting (weights based on terms in topics)? : yes
      - Phrase Extraction from Topics? : yes
      - Tokenizer? :
        I. Patterns which are Tokenized: recognized and discarded constructs of general form "relevant documents consist—contain—include—..."
  - Automatically Built Queries (Routing)
    * Topic Fields Used: title, nationality, narrative, factors, descriptions, concepts
    * Average Computer Time to Build Query (in cpu seconds): .04
    * Method used in Query Construction
      - Terms Selected From
        I. Topics: yes
        II. Only Documents with Relevance Judgments: yes
      - Term Weighting with Weights Based on terms in
        I. Topics: yes
        II. All Training Documents: yes
      - Phrase Extraction from
        I. Topics: yes
        II. Only documents with Relevance Judgments: yes
      - Tokenizer
        I. from Topics: recognized and discarded constructs of general form "relevant documents consist—contain—include—..."

- Searching
  - Search Times
    * Run ID : CrnlEA
    * Computer Time to Search (Average per Query, in CPU seconds): 71
    * Component Times : Initial search (getting top 30 docs) 2.5 seconds.

· Expansion of query by 510 terms in top 30 docs, 1 second. Running expanded query, 36 seconds Keeping track of top 1000 docs, 31 seconds (the bottom hundred docs are constantly churning)

* Run ID : CrnlLA
* Computer Time to Search (Average per Query, in CPU seconds): 189
* Component Times : Initial global Search getting top 1750 docs, 8.5 seconds. Reindexing each top doc, splitting into 200 word overlapping windows,comparing query with each window, adding global sim with best local sim, 181 seconds.
* Run ID : CrnlRR
* Computer Time to Search (Average per Query, in CPU seconds): 43 seconds
* Component Times : Calculation of similarities (average of 394 terms per query), 9 seconds.Keeping track of top 1000 docs, 34 seconds.
* Run ID : CrnlQR
* Computer Time to Search (Average per Query, in CPU seconds): 36
* Component Times : Calculation of similarities (average of 372 terms per query), 11 seconds. Keeping track of top 1000 docs, 25 seconds.Note. Each query was a separate invocation of the SMART.
* Run ID : CrnlVS
* Computer Time to Search (Average per Query, in CPU seconds): 3.2 seconds
* Component Times : Calculations of similarities .08 seconds Keeping track of top 1000 docs, 3.12 seconds
* Run ID : CrnlES
* Computer Time to Search (Average per Query, in CPU seconds): 27.1
* Component Times : Calculations of similarities (average of 525.9 terms per query), 1.3 seconds Keeping track of top 1000 docs, 25.8 seconds (the bottom hundred docs are constantly churning!)

– Machine Searching Methods
  * Vector Space Model? : yes
  * Probabilistic Model? : yes (some queries in CrnlQR)
– Factors in Ranking
  * Term Frequency? : yes
  * Inverse Document Frequency? : yes
  * Other Term Weights? : weights in learning docs for routing. weights in top retrieved docs for CrnlEA
  * Proximity of Terms? : to form adjacent phrase terms
  * Information Theoretic Weights? : yes in CrnlQR
  * Document Length? : yes? (cosine normalization)
– Machine Information
  * Machine Type for TREC Experiment: Sparc 20/51
  * Was the Machine Dedicated or Shared: Dedicated
  * Amount of Hard Disk Storage (in MB): 18 Gbytes
  * Amount of RAM (in MB): 160 Mbytes

    * Clock Rate of CPU (in MHz): 50?
- System Comparisons
    * Amount of "Software Engineering" which went into the Development of the System: 7 Person years?
    * Given appropriate resources
        · Could your system run faster? : yes
        · By how much (estimate)? : Indexing, not much. Retrieval, all runs can be sped up by a factor of 20 using retrieval optimization techniques.