System Summary and Timing
Organization Name: University of Central Florida
List of Run ID's: UCF101 and UCFSJM (each is a routing run).

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Methods Used to build Data Structures
    - Controlled Vocabulary? : Yes.
    - Stemming Algorithm: None
    - Phrase Discovery? : Yes, but it is a manual process when training a filter.
      - Kind of Phrase: Any kind of phrase (a sequence of words) that could be useful.
        For example, "chief executive officer", "due to", "back to committee", "plan that would insure Americans", and "shut down trading".
      - Method Used (statistical, syntactic, other): Manual observation of viewed training text.
    - Manually-Indexed Terms? : Each topic has its own knowledge base which is derived from an Entity Relationship (ER) schema for the topic. For each topic, the knowledge base primarily takes the form of one or more lists (files). There are two types of files. There is a synonym file for each structure component of the ER schema, and there is a domain file for each attribute specified in the ER schema. A phrase (a sequence of words) can be an entry in a domain or synonym file. Different forms of an entry (such as carry, carries, carried,...) are also put in these files. These files are initially built from information found in a dictionary, a thesaurus, or any specialized reference source. Training text is manually viewed to make modifications to these files. The knowledge base for a topic also includes another information (INF) file. The INF file specifies the size of a window for evaluating text, along with the importance of the individual domain and synonym files for determining relevancy of the text in the window.
  - Statistics on Data Structures built from TREC Text
    - Knowledge Bases
      - Run ID : UCF101 and UCFSJM (each is a routing run).
      - Use of Manual Labor
    - Special Routing Structures
      - Run ID : UCF101 and UCFSJM (each is a routing run).
      - Type of Structure: Hash table in memory to store entries in the synonym and domain files of a particular topic filter before beginning input text scan.
      - Total Storage (in MB): Insignificant.
      - Total Computer Time to Build (in hours): A few seconds.
      - Automatic Process? (If not, number of manual hours): Yes.
      - Brief Description of Method: Before a filter starts scanning input text documents, its synonym and domain files are read and each entry is placed in a memory resident hash table.

    * Other Data Structures built from TREC text
- Data Built from Sources Other than the Input Text
    * Internally-built Auxiliary File
        · Domain (independent or specific): Domain specific, a set of synonym and domain files are built for each topic.
        · Type of File (thesaurus, knowledge base, lexicon, etc.): Each file is a list of words or phrases (a sequence of words). A synonym file is constructed for a component of an ER schema. A domain file is constructed for an attribute in an ER schema. Alternate forms of words or phrases are also placed in these files.
        · Total Storage (in MB): For the fifty routing topics, total storage for the synonym and domain files was 214K.
        · Number of Concepts Represented: The concepts represented by a filter's synonym and domain files are ER schema entities, attributes, relationships, roles, subset predicates, specializations, generalizations, and categories.
        · Type of Representation: An Entity Relationship (ER) Schema for a topic.

        · Total Manual Time to Build (in hours): Manually building the synonym and domain files for a single topic ranged from one hour to sixteen hours, the average time was ten hours. However, we did not have enough time to build the best possible files.
        · Use of Manual Labor
            I. Mostly Manually Built using Special Interface: The files were manually built using only an editor. Initially, the files are established using reference material such as a dictionary, a thesaurus, or any specific reference book. The files were later modified after viewing training text. A few special interfaces were used during the training process.
        · Internally-built Auxiliary File
            I. Domain (independent or specific): Domain specific, an information (INF) file. One is built for each topic.
            II. Type of File (thesaurus, knowledge base, lexicon, etc.): The INF file specifies the insertion criteria for a topic's ER schema.
            III. Total Storage (in MB): Small, about 100 bytes each.
            IV. Number of Concepts Represented: The INF file specifies the size of a sliding window (the number of words) used to determine membership in specified combinations of synonym and domain files. The importance of each synonym and domain file is also indicated in the INF file.
            V. Type of Representation: The insertion criteria for an Entity Relationship schema.
            VI. Total Manual Time to Build (in hours): A few seconds to establish one, but an hour or so of wait time to see how good the INF file was for the filter. We did not have enough time to build the best possible INF file for each of the filters.

VII. Use of Manual Labor

A. Mostly Manually Built using Special Interface: An INF file is manually built using only an editor. An INF file is usually modified after viewing training text. In an INF file, the window size and weights of individual synonym and domain files were also modified by observing successive performance evaluations over training text. This was done (when we had time) to obtain optimum performance of a filter over the training text. We did not have enough time to build optimum filters.

· Externally-built Auxiliary File

– Manually Constructed Queries (Routing)

* Topic Fields Used: All.
* Average Time to Build Query (in Minutes): This is the time to sketch an ER schema for a topic (typically, one hour) plus the time to build synonym and domain files for the schema (average is ten hours) plus a minute or so to create the INF file for the topic.
* Type of Query Builder
  · Computer System Expert: The person constructing the synonym and domain files for a topic was an undergraduate student in a Computer Science database course.
* Data Used for Building Query from
  · Only Documents with Relevance Judgments: Yes, if we had time.
  · Other Sources: Hardcopy references (such as a dictionary, a thesaurus, or a specialized reference book) were used. During training, some documents were retrieved that had no definite relevance judgment, so these documents were read and used if the student felt they were relevant.
* Method used in Query Construction
  · Term Weighting? : A weight can be assigned to each synonym file and each domain file of a filter.
  · Boolean Connectors (AND, OR, NOT)? : A form of AND and NOT is used when a combination of synonym and domain files is specified. A form of OR is used when different combinations of synonym and domain files are listed.
  · Proximity Operators? : The sliding window (number of words) to evaluate relevancy.
  · Addition of Terms not Included in Topic? : Yes!
    I. Source of Terms: Any kind of reference material and viewed training text.

• Searching

– Search Times

* Search Times
  · Run ID : UCF101 and UCFSJM (each is a routing run).
  · Computer Time to Search (Average per Query, in CPU seconds): Since each routing query was a true filter that scanned across the entire

document collection, we kept track of wall clock time. For all of Vol. 3, the fastest time was 1 hour and 23 minutes, while the slowest time was 2 hours and 45 minutes. For the SJMN directory, the fastest time was 23 minutes and the slowest time was 28 minutes. The difference in times was determined by the number of filters running on the network at the same time. This ranged from one filter to about eight filters running simultaneously when producing UCF101 and UCFSJM.

- Component Times : No component times, just run the filter across the document collection.

- Machine Searching Methods
  - Machine Searching Methods
    - Boolean Matching? : Somewhat.
    - Free Text Scanning? : Yes.
    - Other: A window (number of words) to view was moved across a document collection and the window was evaluated in regard to words that satisfied the insertion criteria for an Entity Relationship (ER) schema of a topic description. This could be Conceptual Graph Matching.

- Factors in Ranking
  - Factors in Ranking
    - Term Frequency? : Yes.
    - Other Term Weights? : Yes. Each synonym or domain file can be assigned an integer "importance" determined by optimum performance over training text. We did not have enough time to determine optimum numbers.
    - Position in Document? : Yes, sliding window of words to evaluate.
    - Proximity of Terms? : Yes, sliding window of words to evaluate.
    - Other: 1. Number of synonym and domain files for a filter.
      2. Local evaluations (in the window) and a global evaluation of the entire document are used.
      3. Multiple combinations of synonym and domain files are allowed for a filter.

- Machine Information
  - Machine Type for TREC Experiment:
    For training:
    1. The Vol. 1 CD was copied to the hard drive of a PC running Linux (a public domain version of Unix) and functioning as an NFS node.
    2. The Vol. 2 CD was copied to the hard drive of a SPARCserver 690MP (4 processors).
    3. Students ran filters and viewed training text from 32 RISC 6000 machines across a network.

    For UCF101 and UCFSJM runs:
    1. The Vol. 3 CD was copied to the hard drive of a SPARCserver 690MP (4 processors).

2. Most filters were run on the SPARCserver 690MP (a few ran on the RISC 6000 machines).

  * Was the Machine Dedicated or Shared: Shared, except for the NFS node running Linux.
  * Amount of Hard Disk Storage (in MB): We had access to 1000 MB on the NFS node, and 1000 MB on the SPARCserver.
  * Amount of RAM (in MB): 8 MB on each of the 32 RISC 6000 machines. 16 MB on the NFS node. 128 MB on the SPARCserver.
  * Clock Rate of CPU (in MHz): 33 MHz for the NFS node. Not known for the RISC 6000 machines. Not known for the SPARCserver 690MP.

- System Comparisons
  * Amount of "Software Engineering" which went into the Development of the System:
    80 hours: Purchase and install hardware and establish network access.
    160 hours: Design, code, and test the basic filter scanner.
    40 hours: Design, code, and test a few utilities for training.
    40 hours: Figure out the rules for drawing an "atomic" ER diagram.
    100 hours: Establish an ER schema diagram for each of the routing topics.
    500 hours: Establish synonym and domain files for the routing topics for the two submitted runs.

  * Given appropriate resources
    · Could your system run faster? : Yes.
    · By how much (estimate)? : It is a function of how many machines are available for running a filter, and how much traffic the network will tolerate. It might be possible to put a filter on each processor of a machine like the MASPAR, and in four iterations, filter the documents on Vol. 3 in about four minutes.
  * Features the System is Missing that would be beneficial:
    1. A human-computer dialog interface to automate the development of an ER "atomic" schema from a person with a search request.
    2. Access to electronic dictionaries, thesauri, and reference material for initial filter construction from the ER schema.
    3. Utility programs to help train the filters using training documents and relevancy judgments.
    4. An interface for filter modification during interactive queries.

- Significant Areas of System
  * Brief Description of features in your system which you feel impact the system and are not answered by above questions: