System Summary and Timing
Organization Name: Dublin City University
List of Run ID's: DCUNL1 DCUNL2

- Construction of Indices, Knowledge Bases, and other Data Structures
  - Methods Used to build Data Structures
    - Length (in words) of the stopword list: all words in the STOPWORD grammatical category of RUCL
    - Stemming Algorithm:
      - Morphological Analysis: RUCL parser developed as part of SIMPR project
    - Term Weighting: tf*IDF and a combination of lexical labels
    - Phrase Discovery? :
      - Kind of Phrase: yes
      - Method Used (statistical, syntactic, other): syntactic
    - Syntactic Parsing? : yes
    - Word Sense Disambiguation? : no
    - Proper Noun Identification Algorithm? : no
    - Other Techniques for building Data Structures: lexical parsing of text using RUCL parser
  - Statistics on Data Structures built from TREC Text
    - Inverted index
      - Run ID : DCUNL1 DCUNL2
      - Total Storage (in MB): 157
      - Total Computer Time to Build (in hours): 50
      - Automatic Process? (If not, number of manual hours): yes
      - Use of Term Positions? : no
      - Only Single Terms Used? : no
    - Clusters
    - N-grams, Suffix arrays, Signature Files
    - Knowledge Bases
    - Special Routing Structures
    - Other Data Structures built from TREC text
      - Run ID : DCUNL1 DCUNL2
      - Type of Structure: Tree Structured Analytics (tree representations of syntactic structure of clauses)
      - Total Storage (in MB): 672
      - Total Computer Time to Build (in hours): 2200
      - Automatic Process? (If not, number of manual hours): yes
      - Brief Description of Method: lexical parsing of text using RUCL parser. This parsed text is then converted into sets of TSAs

- Query construction

- Automatically Built Queries (Ad-Hoc)
  - Topic Fields Used: narrative and description
  - Method used in Query Construction
    - Term Weighting (weights based on terms in topics)? : tf*IDF and lexical label weighting
  - Phrase Extraction from Topics? : yes
  - Syntactic Parsing of Topics? : yes
  - Other: construction of TSAs from queries to replicate the syntactic structure used in query text

- Searching
  - Search Times
    - Run ID : DCUNL1 DCUNL2
    - Computer Time to Search (Average per Query, in CPU seconds): 1800
    - Component Times :
      - parsing of query : 240
      - presearch using tf*IDF index : 300
      - (threshold of 17
      - TSA matching : 1260
  - Machine Searching Methods
    - Other: our approach is a combination of VSM and tree structure matching where the tree structures are derived from syntax.
  - Factors in Ranking
    - Factors in Ranking
      - Term Frequency? : yes
      - Inverse Document Frequency? : yes
      - Other Term Weights? : yes
      - Syntactic Clues? : yes
      - Proximity of Terms? : yes, by finding the common ancestor in the TSAs to see if two matched terms are in a similar syntactic construct
      - Document Length? : yes, normalise by the number of clauses in the document
  - Machine Information
    - Machine Type for TREC Experiment: Sun Server 690MP
    - Was the Machine Dedicated or Shared: Very Shared
    - Amount of Hard Disk Storage (in MB): 14,000
    - Amount of RAM (in MB): 128
    - Clock Rate of CPU (in MHz): don't know
  - System Comparisons
    - Amount of "Software Engineering" which went into the Development of the System: Excluding the development of the RUCL parser, all software was developed in-house and the software engineering took 4 months with about 15 months of background research and experimentation

    * Given appropriate resources
        · Could your system run faster? : yes ... much
        · By how much (estimate)? : many orders of magnitude ... the system used is a demonstrator prototype with no effort put into optimising the execution speeds at all.
    * Features the System is Missing that would be beneficial: word-word conceptual similarity scoring is where our next efforts will be put

– Significant Areas of System
    * Brief Description of features in your system which you feel impact the system and are not answered by above questions:
        · Our system is implemented as a two-phase operation. A pre-search based on tf*IDF weighting of word baseforms gives us the top 1000 documents and these 1000 documents are then re-ranked using our TSA-based scoring mechanism. This implementation strategy was adopted for efficiency reasons.