

IR-Wire: Gnutella Data Crawler And Analyzer



Database and Information Retrieval Lab
Illinois Institute of Technology
10 W 31st St,
Chicago, IL 60616

Website: <http://ir.iit.edu/~waigen/proj/pirs/>

Wai Gen Yee waigen@ir.iit.edu
Ophir Frieder ophir@ir.iit.edu
Linh Thai Nguyen nguylin@iit.edu
Dongmei Jia jiadong@iit.edu

DOCUMENTATION

I. LimeWire source code modifications:

1. Data logger:
 - a) `core.com.limegroup.gnutella`:
 - *ConnectionManager.java*: increases the number of concurrent connections
 - *FileManager.java*: modifies QRT (set all bits to 1) to log all incoming queries
 - *MessageRouter.java*: Log incoming queries and PONGs. Also, changes the way a leaf node sends out a query (we need to send out sampling queries to more ultra-peers in a random way.)
 - *RouterService.java*: makes *recordAndSendQuery* public in order to send out sampling queries.
 - *BrowseHostHandle.java*: modifies to support shared content crawling.
 - b) `core.datalogger`:
 - *HostCrawlerActivityCallback.java*: shared content crawler.
 - *QuerySamplingActivityCallback.java*: sample the shared content by querying the network.
 - *MySQLMediator.java*: mySQL database interface.
 - *HostIPAddress.java*: host IP address data structure.
 - c) `gui.com.limegroup.gnutella.gui.search`:
 - *SearchMediator.java*: To support query sampling.
 - d) `gui.com.limegroup.gnutella.gui.menu`:
 - *IIRWireMenu.java*: Data logger menu.
 - e) `Lib.messagebundles`:
 - *MessageBundles.properties*: String resource definitions.
2. Ranking functions
 - a) `core.ranker`:
 - *Ranking.java*: Abstract class for all ranking functions.
 - *cosineSim.java*: implements cosine similarity ranking function.
 - *Precision.java*: implements precision ranking function.
 - *TermFrequency.java*: implements term frequency ranking function.
 - *CleanTerm.java*: removes stop words and do stemming.
 - *Stemmer.java*: implements Porter stemming algorithm.
 - *TfIdf.java*: store local statistics
 - b) `gui.com.limegroup.gnutella.gui.search`:

- *SearchTableColumns.java*: Adds new columns for cosine similarity, term frequency, and precision ranking scores.
- *TableLine.java*: To support search results re-ordering by ranking scores.
- *ResultPanel.java*: store search information (the query) in each result line
- *SearchResult.java*: add search information

II. IR-Wire: Data analyzer:

1. GUI:
 - *MainWind.java*: just the main window.
 - *DataAnalyzerPanel.java*: Tabbed panel for analysis results
2. core:
 - *correlationAnalyzer.java*: implements term correlation analysis for queries and shared files.
 - *languageModelAnalyzer.java*: builds and displays language models for queries and shared files.
 - *MySQLMediator.java*: MySQL database interface.
 - *ranker.java*: query and file ranker.
 - *stopWordCollection.java*: builds a list of stop words.
 - *termCorrelation.java*: data structure for term correlation analysis.
 - *termPair.java*: data structure for term correlation analysis.

III. External Libraries

1. mysql-connector-java-3.1.13-bin.jar: to connect to MySQL database.
2. jfreechart-1.0.13.jar: to draw charts.
3. jcommon-1.0.6.jar: used by jfreechart.

IV. Data schema

1. hostdata: store shared content information (one row per each replica on each peer.)
 - host_ID char(48)
 - file_SHA1_value char(42)
 - file_name varchar(255)
 - crawl_time timestamp
2. hostinfo: store peer information, one row per each peer (remove?)
 - host_ID char(48)
 - IP char(15)
 - Port integer
 - Num_shared_files integer
 - Browse_host_support tinyint(3)
 - Last_active_time timestamp
 - Ntimes_crawl integer
3. incomingqueries: store incoming query information (one row per query.)
 - query_GUID char(48)
 - query varchar(100)
 - creation_time timestamp
 - query_type tinyint(3)
(0: no-constraint, 1: audio, 2: video, 3: image, 4: document, 5: application)
4. metadata: store metadata for shared file, one row per each meta tag of a replica (remove?).
 - host_ID char(48)
 - file_SHA1_Value char(42)
 - tag_name char(60)
 - tag_value varchar(255)
5. samplingqueries: records the sent-out sampling queries.
 - query_GUID char(48)