THE ARCHITECTURE DEMONSTRATION SYSTEM

TIPSTER SE/CM

tipster@tipster.org

THE ARCHITECTURE DEMONSTRATION PROGRAM

The TIPSTER demonstration software program shows how the Architecture can meet the TIPSTER goals and requirements. The demonstration :

- ï Supports document detection, information extraction and document management
- ï Allows the interchange of components from different contractors
- ï Uses sharable components interfaced through standard protocols
- ï Allows document detection and information extraction to work together by using standardized linguistic annotations
- ï Supports foreign language text display and processing

The program had two versions, known as the "6-month demo" and the current "12-month demo". The basic object of the program was to showcase the way that different vendor components could work together under the Architecture.

The participants who produced the 12 month demo are:

- ï BBN - text extraction
- ï HNC Software - document detection
- ï Lockheed Martin - text extraction and extraction database
- ï New Mexico State University (NMSU) - document manager and GUI
- ï SRI - text extraction
- ï TRW - document detection output
- ï University of Massachusetts (UMass) - document detection

The demonstration program is not "canned" but allows the various components to operate in a manner similar to an actual application.

A variety of document collections in English, Spanish and Arabic are available under the Document Manager to support the various components. All components use the common Document Manager and common viewers for collection lists, document lists, documents, Detection Needs and Annotations in the Graphical Users Interface(GUI). The GUI is not part of the Architecture but the use of only one GUI illustrates the standardization of component outputs and is another example of the sharing of common facilities by diverse components.

DOCUMENT DETECTION

Document Detection includes the selection of documents from a corpus, with the output rated by relevance, and the routing of documents to users, based upon Detection Need profiles. In the first case a Detection Need is passed against the entire corpus. In the routing case the corpus is passed against a group of Detection Needs. Routing is not supported in the demonstration.

The Detection Need format supports natural language queries as well as Boolean keywords and fuzzy queries. A unique capability of specifying the Detection Need is the ability to accept sample documents that represent the relevant documents or sample documents which represent the non-relevant documents (so called Query-by-Example). Even TREC-type topics can be used as a Detection Need.

A document Collection may be selected, a Detection Need created or selected from a library and these items sent to the detection engine of choice, HNC Software or UMass, each of which use entirely different algorithms for detection.

The resulting list of documents are rank ordered by best match. After examining the results from the query process the Detection Need

may be modified for a new run. After the results of the previous run are examined, sample relevant or non-relevant documents are typically selected from the run and used to modify a Detection Need so as to improve the query.

Each of the detection engines uses entirely different methods to obtain their output, but both can accept the same Detection Need and both produce TIPSTER compatible output.

The results of a Detection Need may be viewed as a list of relevant documents. The specific document text may also be viewed and in some cases the text that contributed to the document being selected as relevant can be highlighted.

INFORMATION EXTRACTION

Information Extraction (IE) is the identification and output of specific types of data elements and relationships, from free text. At the basic level of name-spotting, the typical elements identified are person and organization names, cities, countries, dates, and numeric expressions such as monetary figures. These data elements are highlighted in color in the documents on screen. Relationships represent more complex extraction techniques. This processing necessitates linking; e.g., person P with company X, and showing that company X has entered into a joint venture agreement with company Y to manufacture a product R through X's subsidiary, company Z. The results of this processing are output in a relational template.

In the Demonstration, name-spotting capability is shown by the SRI, BBN, and LM systems. SRI's FASTUS system can also recognize tables embedded in text.

Annotations are the mechanism for collecting extracted information from a document including the actual data, the type of data and any relationships for the data. All the extraction components produce annotations in the same structure and format. Thus, annotations with extracted information or linguistic information may easily be passed between components that need the information and a common viewer may be used to examine the annotations on a document, regardless of which component did the extraction. Typically, extracted information is used to build domain databases.

BBN and LM both provide template viewers for displaying the extraction of relationships that are comprised of multiple annotations, each of which can then be examined in detail. To demonstrate how extraction technology can be integrated with that of detection, the LM template viewer also provides the means for designating any of the extraction output and generating automatically a query which incorporates the selected output as constituent elements of the query. Transparent to the user, the query runs in SQL against a hand-crafted Oracle database of joint venture articles and retrieves relevant documents.

The current version of the demonstration took four months to build and operates on a SUN/UNIX system.