The Semantic Analysis Workbench (SAW): Towards a Framework for Knowledge Gathering and Synthesis

Anthony Levas, Eric Brown, J. William Murdock, and David Ferrucci
IBM TJ Watson Research Center
Yorktown Heights, NY 10598, USA
{levas, ewb, murdockj, ferrucci}@us.ibm.com

Abstract
We describe a tool called the Semantic Analysis Workbench (SAW) that integrates a broad set of analysis and search functions in support of the Unstructured Information Management (UIM) lifecycle. We discuss the SAW and articulate some initial thoughts for an underlying framework that supports Knowledge Gathering and Synthesis (KoGS) tasks.

1. Introduction
Unstructured information is the fastest growing source of current, high value information available. However, extracting relevant content from large volumes of multimodal data and representing it in forms required by search and mining applications that can be used by Intelligence Analysts is a challenging problem. Analysts need tools and techniques to help focus data collections to manageable sets containing relevant content. Lightweight analytics, such as that required by keyword search, may be appropriate for large sets of data. Deep analytics, such as that required for fact and deductive search, need to work on smaller volumes of information to be effective. While incrementally seeking a rich set of facts and justifying evidence, analysts need to prudently apply analysis to unstructured information at different phases of the knowledge gathering and synthesis task.

In this paper we describe a tool called the Semantic Analysis Workbench (SAW). It was built using the UIMA SDK (SDK 2004), a Java implementation of the IBM Unstructured Information Management Architecture (UIMA) (D. Ferrucci and A. Lally 2004 and 2004b). The SAW integrates a set of analysis and search functions to support development of analytics, configuring and running analysis on collections of data, and exploring the results using different search paradigms, each requiring incrementally complex analysis; these include keyword and semantic search as well as fact search for viewing entities and their relations. We will describe the SAW in the next section. This is followed by the conclusion section, where we discuss some thoughts regarding the process of knowledge gathering and synthesis and an underlying framework to support it.

2. The SAW
The SAW is an application that integrates a broad set of tools that are used throughout the development process of UIM applications. The SAW supports three orthogonal activities for managing this life cycle: 1) Develop Analysis, 2) Configure and Run Analysis and 3) Explore Results.

To support the Develop Analysis phase, the SAW launches the Java Eclipse platform (Eclipse 2003) loaded with the UIMA SDK, along with a set of specialized plugins. Developers use this IDE to write and test analytics that are used in UIM applications.

The Configure and Run Analysis phase is supported by providing access to the Collection Processing Engine (CPE) editor tool, a graphical tool for building CPEs. This tool allows analysts to select, parameterize, and sequence components, as well as run the analysis. The CPE is used to analyze a set of unstructured artifacts that have been aggregated into a collection, and as a result produces structured information repositories.

The Explore Results phase is supported by a rich set of search capabilities that are used to query the structured information derived from the analysis phase. The SAW provides four major search capabilities, each contained in its own tabbed pane in our graphical user interface. They are: Keyword Search, Semantic Search, Entity Search and Fact Search.

Keyword Search provides traditional free text search, where the user enters a “bag of words” query and the search engine uses the index built during the analysis phase to process the query. Similar to a typical web search, the keyword search engine returns a hit-list of documents rank ordered by relevance.

The Semantic Search engine goes beyond keyword search by exploiting any semantic analysis performed during the analysis phase. Semantic analysis, such as named entity detection, produces annotations over the document. An annotation associates a semantic type and arbitrary properties with a specific span of text. For example, a named entity detector will annotate the text “John F. Kennedy” with the semantic type PERSON. The Semantic Search engine provides an enhanced query language, called XML Fragments (Carmel et. al. 2002), where the user can incorporate semantic types into the query. In particular, the user can specify that query terms must appear within the span of a particular semantic type. XML Fragments also supports nesting of semantic types, enabling the user to construct complex, structured queries.

The SAW provides a tool that allows users to graphically compose an XML Fragment query by selecting se-
Semantic types presented in a list derived from the set of all known types found in the collection. In addition, a user can build nested structures using this editor to compose complex queries.

Semantic Search provides a very powerful search capability returning artifacts based on semantic category. For example, this capability can be used to focus or narrow a search e.g., to disambiguate between artifacts referring to the Person Kennedy, the Airport Kennedy, or the Facility Kennedy (hotel). In other situations, it can be used to cast a broad search query, returning all artifacts that contain a particular tag. For example, `<PERSON>` would return all documents that include at least one Person annotation.

Entity Search and Fact Search operate at a different level. Specialized analytics, in this case, have produced annotations at the Referent Layer. This level uses annotations from prior analytics to produce structured information for the entities and relations discovered in the artifacts undergoing analysis. Entities can be thought of as individual things that exist in the data - for example, the specific person John F. Kennedy, or the specific country, the United States.

Coreference analysis relates mentions in a document that have different forms for the same entity. For example, IBM, and International Business Machines should resolve to the same entity. Pronominal reference can also be resolved. For example, in "John went home. He took the bus", both “John” and “he” refer to the same entity, requiring these facts be associated with the specific entity John. In addition, coreference must span documents, enabling facts to be collected for an entity across many documents. During the analysis phase, we extract entities and relations and encode their structured information in a repository called the Extracted Knowledge Data Base (EKDB). The EKDB is a relational database and can be queried using a standard SQL API. The graphical user interface provided by the SAW, however, shields the user from this level and allows queries to be easily created without requiring knowledge of SQL.

3. Conclusion
The SAW provides a powerful set of capabilities applicable to the UIM lifecycle. It has proven to be an effective tool for developers of UIM applications as well as for those interested in just viewing the results of analysis. We have used the SAW extensively as a tool for demonstrating UIMA and the capabilities of the underlying analytics as well as a pedagogical tool in many tutorials. See (Levas et al. 2005) for a description of an application using the SAW. The SAW is our first prototype application integrating a broad set of tools that support the UIM lifecycle. This work has motivated our thinking about the process an analyst undergoes during an investigation, which we call Knowledge Gathering and Synthesis. We define this process as: An incremental and iterative exploration of structured and unstructured information to discover, gather and integrate knowledge related to a particular inquiry.

Our assumption is that analysts engaged in complex investigations require systems that integrate different Analytics, Search Techniques and Results Management tools within a Software Framework that supports the process of Knowledge Gathering and Synthesis (KoGS). This framework bridges the UIMA Layer and the Application Layer above, enabling the rapid development and integration of a wide array of UIM applications.

Our future work in developing the KoGS framework will focus on a few key areas. Management of results of analytics, such as Semantic Search indexes and KnowledgeBases has emerged as a key requirement. Issues, such as: naming and storing; merging new information; removing unwanted information; declaring the type systems that the analytics and other components used in creation; re-useable graphics widgets for visualization; and a host of other concerns need to be addressed to effectively manage analysis results in UIM applications. Management of collections of artifacts is also very important. Similar issues, such as: creating artifact collections, naming them, merging them, pruning them, etc., are especially important. Finally, managing the analysis session has also emerged as an important requirement. For example, allowing an analyst to keep and browse a history of his activity, fruitful results and provenance information, as well as, dead ends encountered, can help him in this process, and with sharing this information with colleagues. The three topics above are just a few of the key areas that we will begin to explore as we develop the KoGS framework.

References
D. Ferrucci and A. Lally 2004a. Building an example application with the Unstructured Information Management Architecture. IBM Systems Journal 43, No. 3, 455-475.
D. Ferrucci and A. Lally 2004b. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10, No. 3-4, 327-348.