Semantically-Enhanced Information Extraction

Hisham Assal	John Seng	Franz Kurfess	Emily Schwarz	Kym Pohl
CAD Research Center	Computer Science	Computer Science	Computer Science	Tapestry Solutions, Inc.
Cal Poly University	Cal Poly University	Cal Poly University	Cal Poly University	*A Boeing Company*
San Luis Obispo, CA	San Luis Obispo, CA	San Luis Obispo, CA	San Luis Obispo, CA	San Luis Obispo, CA
hassal@gmail.com	jseng@calpoly.edu	fkurfess@calpoly.edu	eschwarz@calpoly.edu	kpohl@tapestrysolutions.com

Abstract—Information Extraction using Natural Language Processing (NLP) produces entities along with some of the relationships that may exist among them. To be semantically useful, however, such discrete extractions must be put into context through some form of intelligent analysis. This paper[1,2] offers a two-part architecture that employs the statistical methods of traditional NLP to extract discrete information elements in a relatively domain-agnostic manner, which are then injected into an inference-enabled environment where they can be semantically analyzed. Within this semantic environment, extractions are woven into the contextual fabric of a user-provided, domain-centric ontology where users together with user-provided logic can analyze these extractions within a more contextually complete picture. Our demonstration system infers the possibility of a terrorist plot by extracting key events and relationships from a collection of news articles and intelligence reports.

Table of Contents

Introduction

Shortly after the appearance of a large collection of documents related to the war in Afghanistan on the Wikileaks Web site in July 2010 [1], the US government combed through the documents in order to seek out passages that might endanger people by inadvertently disclosing information about their identities:

Mr. Morrell said the Pentagon had formed a team of 80 analysts from the military and the F.B.I. who are working around the clock to vet the documents for damaging information. So far the team, which is expected to increase to about 125 people in the coming days, has conducted about 400 "key word" searches through the 77,000 disclosed documents. When those searches turn up information, Mr. Morrell said, it is set aside for further analysis. After this initial review is completed, the Pentagon will conduct a separate "page by page, word by word" review of each and every document, he said. [2]

With a collection of documents that large, keyword searches are a reasonable first step, and the sensitivity of the information necessitates a careful, albeit time-consuming analysis. Computers, however, can offer more sophisticated support for information extraction and intelligence analysis than keyword searches, and hopefully some branches of the U.S. government have access to such capabilities, enabling a more effective and more efficient analysis and review of such documents.

In this paper, we describe a framework that is intended to provide such support to intelligence analysts and knowledge workers that perform similar tasks.[3] The basis is a community of agents organized as a service-oriented architecture (SOA); see Figure 3. Agents perform and utilize services, including some provided by outside service agencies. This community of collaborative agents shares a common objective, but the individual agents have their own capabilities and the autonomy to make their own decisions. Each agent offers a set of services, and in turn may rely on services provided by other agents.

One set of services is responsible for document access, which includes Web crawlers, internal search services, and possibly specialized services to access proprietary repositories. Another set of services performs information extraction. The main task here is to perform natural language analysis on the documents, and the extraction of named entities such as persons, places, and events along

[1] 978-1-4244-7351-9/11/$26.00 ©2011 IEEE.

[2] IEEEAC paper #1121

[3] It is indeed our plan to use it on document collections similar to the one mentioned above. However, at the time of writing, our implementation is not robust enough to deal with a collection of that size.

with the relationships that connect them. Then there is a set of services related to the construction, maintenance, and use of the domain model. These services provide access to various ontologies, relate specific instances (facts) to abstract concepts in the ontologies, and to reasoners that apply rules to the knowledge contained in the domain model. The final set of services incorporates direct access to specific tools, such as lookup services using Google and Wikipedia for checking the plausibility of easily verifiable statements.

Background and Previous Work

For a better understanding of the problems involved in information extraction, one should examine the basic workings of a search engine, such as Google or Bing. They use Web crawlers to gather documents, then create a massive index of those documents by noting which words occur in which documents, and answer queries by identifying documents that contain the keywords identified in the query. Since the result may be a large set of documents, an important step is the ranking of the documents in order to present the ones that are most likely to contain the answer to the user first. Google's initial success relied on a smart way of ranking documents and delivering better results. While search engines provide invaluable services to users, logical analysis reveals several significant limitations:

(1) String-based indexing: Matching documents to queries relies on the co-occurrence of words as sequences of letters in the query and in the document. This can be improved through techniques like stemming and query expansion, but is different from the way humans automatically try to associate a word with its intended meaning.

(2) Content-based/concept-based search: In conventional search engines, there is no explicit consideration of the meaning of words. While some approaches include techniques like query expansions based on thesauri, the main emphasis is on statistical correlations between text strings and documents.

(3) Task context: Users typically perform a query within a given context, such as writing this paper. Search engines have no awareness of the user's context or task model, which could be used to narrow the focus of the search on a particular domain. Clearly there are situations where this separation between the user context and the search engine is appropriate, and the user should be in control of how and when such context information is used.

(4) Interactive use: Typically, search engines are used in a "batch" mode, where the user types a word or phrase into a search box. In most cases, finding an interesting result in the list presented by the search engine satisfies the immediate need of the user, and they have no reason for additional interactions. In some situations, however, users are motivated to pursue a series of interactions with a search engine, typically with the hope of getting better results. This interaction also provides relevance feedback to the search engine, which can be used to improve subsequent results to similar queries.

To overcome these limitations, our system combines methods from Natural Language Processing (NLP), Information Extraction (IE), Knowledge Representation (KR), and Domain Modeling (DM) with a service-based architecture that utilizes agents as providers of services that implement some of the above methods. In contrast to the generic population of Web users that constitute the constituencies of search engines, our approach is intended for users that are strongly motivated to find additional information within a specific domain, and frequently while working on a particular task. In addition, many of them will be members of organizations or communities that have an inherent interest in collaboratively enhancing the collective body of knowledge. Thus we can assume that there exists a domain model (or some users are motivated to construct one), users will interact with a tool that offers support for their tasks and simultaneously enhances the capabilities of their community, and they are willing to share context information about their activities and tasks for search and knowledge organization purposes.

Natural language processing

Natural language processing (NLP) refers to the use of computational methods to analyze and process spoken or written statements in a language commonly used by humans. Such methods are applied from different angles. At the *syntactic level*, grammatical rules are used to determine the basic building blocks of text, such as sentences, words, and the roles they play in a given piece of text. At the *semantic level*, the meaning of words, phrases, sentences and documents is determined. At the *pragmatic level*, the context is taken into account as well to determine the most suitable interpretation. Syntactic analysis is relatively straightforward from a computational perspective, but not sufficient to determine the meaning of a text fragment or document; ambiguity, for example, can drastically change the information conveyed in a sentence. Semantic analysis relies on a common interpretation between the creator (writer) and the consumer (reader) of a document. For humans, this is usually a subconscious, semi-automatic consideration of the intended meaning of a statement. It could simply be that they speak the same language, or that they have a mutual awareness of their respective contexts. Computers operate mostly on the syntactic level, however, and have difficulties devising the meaning of words and phrases. One approach to establish a common interpretation relies on ontologies as frameworks that define the core terminology in a domain and specify the relationships between words. Contextual aspects can be explicitly specified (e.g. through rules), incorporated into a domain-specific ontology, or derived from additional information about the documents and how they are used.

Statistical approaches in NLP can overcome this interpretation problem to some degree, and are sometimes combined with the structural analysis methods that rely on rules specifying the grammar of the language. In the NLP area, our approach relies on the use of existing frameworks and toolkits [3][4][5][6][7]. Due to the service- and agent-oriented architecture, it is relatively straightforward to use multiple tools, either in parallel, or by switching among them.

Information Extraction

In our context, information extraction (IE) refers to the use of computational methods to identify relevant pieces of information in documents generated for human use, and convert this information into a representation suitable for computer-based storage and processing [8]. IE is often implicitly constrained to text-based documents, although in principle it can be applied to other types such as images, videos or audio recordings as well. For IE, the goal is to identify meaningful chunks of information, which requires the selection of relevant pieces of texts (words or phrases), and the conversion into a computer-suitable format. Since natural language is ambiguous, redundant, and contextual, the task of identifying and extracting relevant pieces of information is very challenging for computers.

Computer-based IE and NLP methods can identify the occurrence of particular text pieces (words, phrases) in documents, allowing the analysis of simple statements about entities, events, and actions, and the comparison of identified entities against other documents. Such systems have been pursued in research projects such as the Open Information Extraction approach at University of Washington [9][10], but are also in use in commercial system such as the Calais approach used by the Reuters news agency [12].

Domain Modeling

Our system aims at the extraction of meaningful pieces of information from wide sets of documents, their integration into a coherent framework, and the derivation of new knowledge. One critical assumption is the existence of such a coherent framework for the domain under consideration. An ontology is a formalized representation of such a framework, and serves multiple purposes in our context. First, it makes explicit the knowledge of humans about the domain. Second, it ensures that the interpretation of critical terms is consistent within the group or organization that utilizes it. Third, it spells out important relationships between those terms. Associated with an ontology can be a set of axioms, which capture the very basic, generally accepted statements about a domain. The flexible use of relations in ontologies allows dynamic, multiple classification of entities. While these properties of ontologies already allow for fairly powerful reasoning, our system also incorporates components for reasoning that are external to the ontology. An overview of the current status of combining information extraction and ontologies is given in [8].

On the NLP side, ontologies are the vehicle to provide a semantic framework for the interpretation of sentences and documents, enabling the conversion of statements available in natural language into a representation suitable for computers. For the IE task, an ontology helps in deciding which pieces of information may be relevant, and how they are incorporated into the already existing knowledge repository. The combination of axioms and a flexible hierarchical structure provides a strong basis for reasoning and analysis of the information captured in the repository. Ontologies also have a natural visual representation as a graph where nodes represent concepts and links relationships between concepts, and thus serve as a powerful information retrieval method by following interesting relationships.

Ontologies provide substantial support for several aspects of our system, such as the explicit representation of domain knowledge, interpretation of text, the analysis of documents, and the identification and retrieval of stored information. However, they are difficult and cumbersome to build, may not be available for some areas of interest, and do not capture the full understanding that humans have. The use of ontologies can also become computationally very expensive [11].

The main purpose of our system is to augment the capabilities of human analysts for dealing with large collections of knowledge and information. It incorporates information retrieval from text documents found on the Web and in proprietary repositories. Through integration with existing domain models in the form of ontologies, analysts can utilize their conceptual models and task contexts as a foundation, and enhance this with additional information identified by our system. It is designed as a modular framework using a service-based, agent architecture, with the explicit goal of facilitating changes in domain models and services with moderate effort.

ONTOLOGY-BASED INFORMATION EXTRACTION

There are a number of implementation differences that distinguish previous ontology-related information extraction systems from each other. We describe the difference in implementations along four dimensions as categorized by [8]: information extraction implementation, ontology usage, ontology extraction specificity, and natural language data source.

The first and probably most significant variation in implementation is how the information extraction is performed in a system. Information extraction itself is a developing field and can be performed using a combination of techniques.

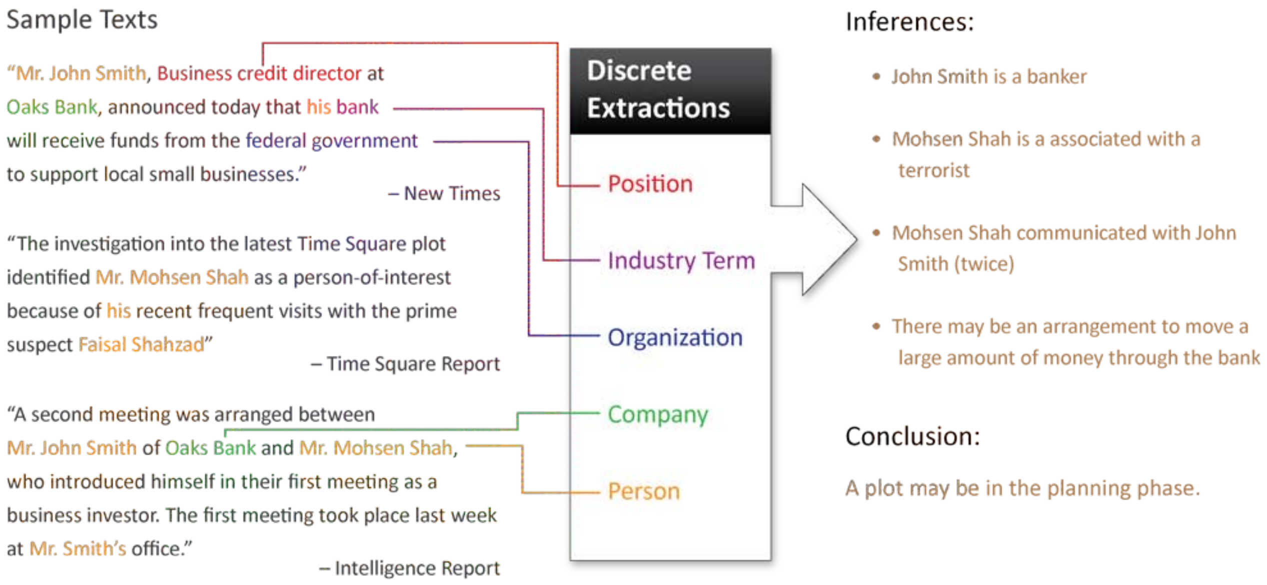

Sample Texts
"Mr. John Smith, Business credit director at Oaks Bank, announced today that his bank will receive funds from the federal government to support local small businesses."
– New Times
"The investigation into the latest Time Square plot identified Mr. Mohsen Shah as a person-of-interest because of his recent frequent visits with the prime suspect Faisal Shahzad"
– Time Square Report
"A second meeting was arranged between Mr. John Smith of Oaks Bank and Mr. Mohsen Shah, who introduced himself in their first meeting as a business investor. The first meeting took place last week at Mr. Smith's office."
– Intelligence Report
Discrete Extractions
Position
Industry Term
Organization
Company
Person
Inferences:
John Smith is a banker
Mohsen Shah is a associated with a terrorist
Mohsen Shah communicated with John Smith (twice)
There may be an arrangement to move a large amount of money through the bank
Conclusion:
A plot may be in the planning phase.

Figure 1 – Illustration of Functional Objectives

The first information extraction technique is to use regular expressions to match phrases in natural language text. These regular expressions are often constructed by a domain expert to perform matches on phrases as they appear in actual text. This approach is tedious, but can often yield high quality results. Another information extraction technique is that of using gazetteer lists. A gazetteer list is a list of known terms and phrases as they exactly appear in text. When text contains a named entity that matches an element in the gazetteer list, then the named entity is extracted. A third approach is to use a machine learning classifier to classify natural language text as relevant information. This approach uses training data (commonly human annotated text) to train a machine learning classifier to learn how information appears in sentences based on some feature set (e.g. part of speech tags, word position, or capitalization).

The second of four implementation differences is how the ontology is used in the system. Some systems use the ontology as user input which a human has pre-defined. This assumes all extracted information items must fit into some portion of the defined ontology. Another approach is to have the system dynamically define the ontology as it processes the natural language input. Such a system would create new objects in the ontology as they are identified at run-time.

The third implementation difference is what portion of an ontology a system can extract. An ontology information extraction system can potentially extract classes, properties of classes, and relationships between classes. Ontology-based information extraction systems can vary on the level of details for a class that is extracted.

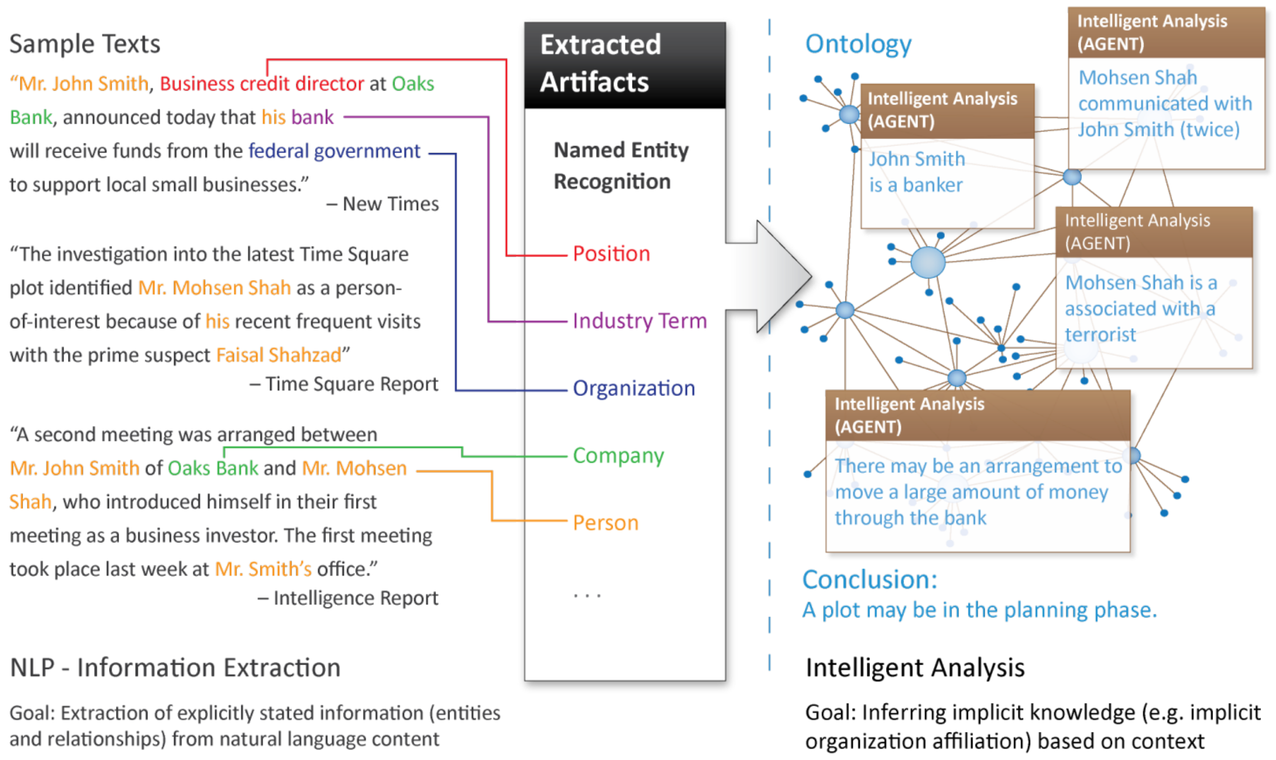
The final variation among information extraction systems is in the source of the natural language data that is processed by the systems. Some systems will use a text source available from a Web site, while other systems require the text to exist in a particular file format.

Research in the information extraction field has been motivated in the past by two research competitions. The Message Understanding Conference [MUC] was a DARPA sponsored event held from 1991-1997. The event required participants to demonstrate systems performing various information extraction tasks. The Automatic Content Evaluation [ACE] program was a NIST sponsored event held from 1999-2008.

In terms of the tools that are available, there are several toolkits that target the development of natural language processing applications [3][4][5][6][7]. Two commonly utilized frameworks are GATE (General Architecture for Text Engineering) and UIMA (Unstructured Information Management Architecture). These frameworks provide services and workflows which simplify the construction of NLP applications. For our work we utilize the UIMA architecture. This framework provides facilities such as annotations, chaining of text-level annotators, and an overall modular development environment.

General Approach

The objective of this research is to offer an environment for collecting and analyzing large volumes of information from online sources, as well as private document repositories (Figure 1). The analysis of text documents has two aspects: information extraction and intelligent analysis. Information extraction from natural language text is performed using existing general NLP tools, which handle the Named Entity Recognition (NER) and Parts Of Speech (POS) aspects of document analysis, combined with special purpose rule-based extraction to assist in identifying domain specific



"Mr. John Smith, Business credit director at Oaks Bank, announced today that his bank will receive funds from the federal government to support local small businesses." – New Times

"The investigation into the latest Time Square plot identified Mr. Mohsen Shah as a person-of-interest because of his recent frequent visits with the prime suspect Faisal Shahzad" – Time Square Report

"A second meeting was arranged between Mr. John Smith of Oaks Bank and Mr. Mohsen Shah, who introduced himself in their first meeting as a business investor. The first meeting took place last week at Mr. Smith's office." – Intelligence Report

Goal: Extraction of explicitly stated information (entities and relationships) from natural language content

Mohsen Shah communicated with John Smith (twice)

John Smith is a banker

Mohsen Shah is a associated with a terrorist

There may be an arrangement to move a large amount of money through the bank

A plot may be in the planning phase.

Goal: Inferring implicit knowledge (e.g. implicit organization affiliation) based on context

Figure 2 - Applied Technologies

relationships. The intelligent analysis aspect is based on a domain ontology which describes the complex relationships that exist in that domain and some of the basic rules of inference, which describe the necessary conditions for classification of objects and relationships. We emphasize the role of the user in this approach to provide assistance in interpreting the extracted information, review the automated inference and make any changes to enhance the quality of information. Applying the selected technologies to these objectives yields a hybrid solution that partners traditional NLP techniques with inference-rich intelligent analysis as shown in Figure 2. The design of this system is based on a Service-Oriented Architecture (SOA), as shown in Figure 3. All of the functional components are implemented as Web services and hosted on an application server. The decision to architect the information extraction capability in a service-oriented manner was twofold. First, the extraction of information appeared to be an activity that would be invoked by external components as a part of some larger business process and done by a wide variety of users and on a potentially repetitive basis. As such, it appeared appropriate to deploy the information extraction capability as a whole in the form of a discoverable and subsequently invocable service. Second, since the information extraction capability is comprised of a number of sub capabilities each of which may be a candidate for replacement with newer, more capable components, it seemed advantageous to carry this service-oriented concept into the internal architecture as well. It was anticipated that doing this would produce a test bed-like environment where internal capabilities were substantially decoupled and primed for potential replacement or partnering with additional, complementary capabilities.

The current service implementation includes the following:

(1) Web crawler service to access sources on the Internet and retrieve articles and other material from Web sites, clean it up (e.g. remove HTML tags) and store it in the document repository.

(2) Feature extraction service to process the textual information, extract objects of interest, as defined by the ontology, and classify some of the extracted objects.

(3) Semantic model service to provide ontology access and manipulation functionality. The Web Ontology Language (OWL) [13] is selected as the paradigm to build the context model(s) applicable to particular target domain(s). OWL supports dynamic, multiple classification, which is important in order to describe a person, for example, as both a 'banker' and a 'terrorist supporter'. It also supports decidable logic, which makes the inference possible. The semantic model service utilizes a reasoner, which is an embedded inference engine that operates over embedded and externalized domain logic. For persistence of model objects we use an RDF [14] triple store.

(4) Tools service to provide additional capabilities to help the user interpret the extracted information and check

the validity of the automated inference. The tools allow for more complex extractions, such as multi-term NER. Tools also help refine and aid in the classification of extracted objects, using encyclopedic services (e.g. Wikipedia).

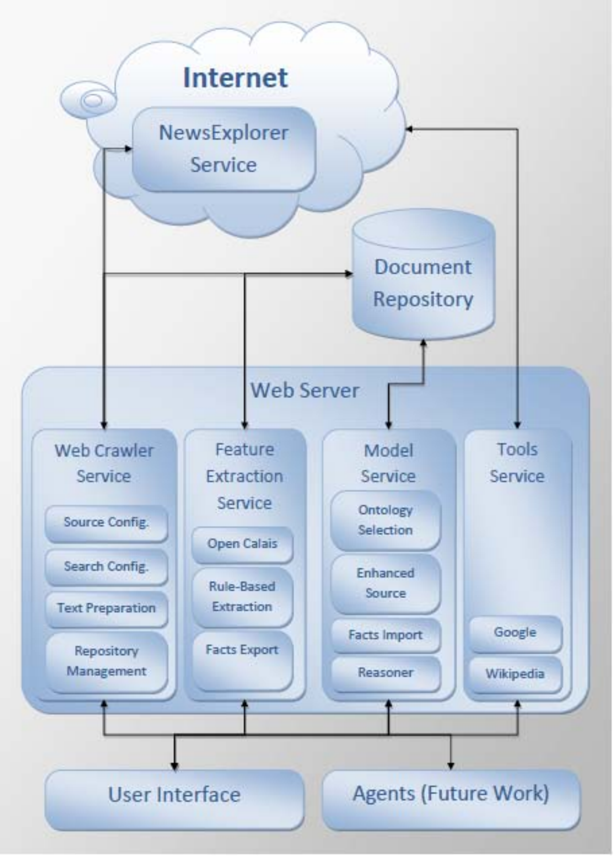


Figure 3 - Conceptual Architecture

In addition to the Web services, there is a document repository, which holds the raw textual information as well as the results of NLP extractions. The repository is accessible to all the services, as well as the user interface.

The concept of the user interface design is to allow the user to be involved at any step of the extraction and analysis process. It is important to have the user input and integrate it into the intermediate results. The user can enhance the quality of extracted information by merging names with different spellings, identifying organization types, adding entities and relationships that may have been missed by the extraction module, or remove redundant information. As was evident in the MUC and ACE series of research initiatives, the accuracy of information extraction systems can still benefit from human user interaction.

The user interface provides tools for the user to access the services, configure them to perform according to the task at hand and provide input when the user decides to intervene.

The UI is Web-based, so that it can be run from any browser-enabled location and support collaboration. The UI offers support for configuration of information sources, setting of various selection criteria, user-initiated extraction of basic (i.e., explicit) concepts, and user-initiated exporting of NLP extractions into the semantic model. The user can also edit the system configuration elements, such as the extraction rules and classification lists.

The reasoning process is triggered by the user and can be configured easily on the UI. The user can select an ontology for reasoning and can load multiple ontologies to support complex inference.

The presentation of the inference results can be done in a tabular form as well as on a graph, depicting the concepts and their relationships. The results are editable and in the final design can be integrated into the on-going inference process.

The UI also offers access to a set of services that provide tools to enhance the interpretation of the extracted information.

The design of this system emphasizes partnership between the human user and the automated tools. The system provides support for activities performed by human users. Therefore, the system has to provide the user with visibility into its automated activities, and offer the user the ability to amend, correct, or enhance the results produced by the tools.

USE CASE OVERVIEW

We present a use case here to illustrate the benefits of partnering the traditional NLP capabilities with ontology-based inference capability. The use case supports intelligence analysis of terrorist activities based on information extracted from news sources as well as intelligence reports. The assumption here is that different pieces of information appear in different documents from multiple sources using related but not identical terms, and that their combination yields an interesting or important new piece of information which can be woven into an evolving contextual fabric. Information may be obtained from a number of sources, with overlapping data items, e.g. a person or an organization may be mentioned in multiple news articles. All the statements are stored in the repository, including any redundant information. When statements are imported into the ontology as facts, a check on information is performed against the current state of the ontology. Information about the same object (person, organization, location, etc.) is merged together to ensure information integrity.

The use case examines information about people, organizations, and communication. The relationships among these three types of objects are also examined, e.g. person's membership in organization, communication between two people, etc.

The scenario supported in the use case is as follows:

Purpose:

(1) To identify the possibility that a suspected terrorist group (or individual) is planning an attack

Explicitly extracted information:

(1) Membership in a known (or suspected) terrorist group

(2) A person employed by a financial institution

(3) A meeting, taking place between the members of the terrorist group and the banker

Inference:

(1) The terrorist group may be planning to transfer a large amount of money through the banker

(2) A terrorist plot may be in the making

This simple use case illustrates the extraction of information from multiple sources (the person's membership in the terrorist group may come from one source, while the meeting between this person and the banker may come from another). It also illustrates the use of an ontology to build a context for the extracted information and the use of the ontology inference capability to create new information.

NLP ENVIRONMENT

This section presents the implementation and design of the natural language processing server portion of the solution. The NLP service suite includes the Web crawler service and the feature extraction service, as shown in Figure 3. The NLP component consists of a combination of very general external Web resources as well as domain-specific rule-based extraction techniques. By utilizing both external Web resources and features tailored to a unique domain, we are able to extract natural language features which may be missed by a more broad-based approach while maintaining an acceptable level of generality for application across a diverse set of potential domains.

From a high-level perspective, it is the job of the NLP component to download text content from the Internet and extract relevant information from key phrases present in the text. The extracted semantic relationships, referred to as *assertions*, are then injected into the semantic analysis component of the solution. As a suite of web services, the NLP environment is directly invocable by the user via the user interface.

The following subsections outline the structure of the NLP service suite. The next subsection describes what criteria are used to acquire news articles for processing. Subsequent subsections cover: the task of cleaning the HTML for a Web article, how natural language features are extracted from the system, and the network implementation of the NLP server suite along with a description of the exposed API.

Article Acquisition

In a high-level view of the system, the NLP portion takes as input natural language text and produces discrete assertions regarding the relevant domain. Such natural language input may come from a variety of online sources: major news Web sites, online forums, and blogs. In the current implementation, the primary source of natural language content is focused on news articles gathered from the Internet. News articles are written for readers who may not have extensive background knowledge on a topic and therefore the articles will often list people's full name and titles. This fact is advantageous for detecting the political and corporate positions of various people.

We use an online news article collection Web site NewsExplorer [15] developed by the European Commission's Joint Research Centre. The Web site performs statistical analysis of various news articles that are released throughout the day. Some sample features of the NewsExplorer Web site are the ability to cluster articles by common terms, country of origin, and primary article topic.

For this work, we focus on a subset of the topics that NewsExplorer supports as not all subtopics are relevant to the described use case. Our graphical application currently allows downloading of articles that are categorized under one of the following topics: Security, TerroristAttack, Conflict, Energy, and AlternativeEnergy.

Article Cleanup

The NewsExplorer Web site provides Web links for each of the articles that it tracks. In order to download an article, the article itself must be purged of any HTML tags and content not relevant to the news article. For this task, we use the HTML cleaning functionality of the AlchemyAPI NLP Web service [16]. Given a Web link of an article, this service returns the raw text of the article void of any HTML markup tags or any other advertising content. Once an article has been cleaned, a local copy is cached for future use.

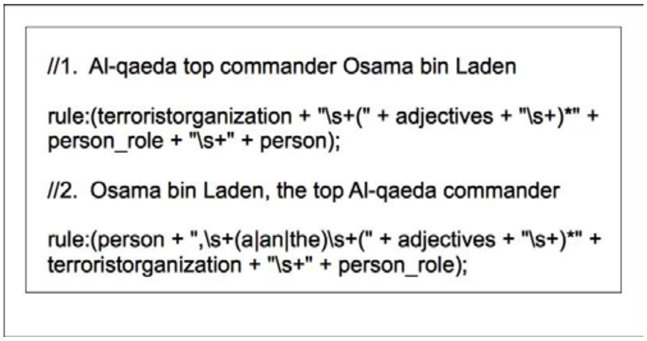
Natural Language Feature Extraction

Although architected to integrate with potentially any variety of supplementary extraction capabilities, the current implementation utilizes two methods of extracting information from natural language text: OpenCalais and our own rule-based technique.

OpenCalais is an information extraction service provided by Thomson Reuters [12]. The service takes natural language text and returns a document with various labeled entities in an RDF format [17].

For our current implementation, we use only a subset of the entities that are detected by OpenCalais. The entities that are stored are *PersonCareer (a person's position with an organization)*, *PersonCommunication (communication between two people)*, *PersonRelation (type of relationship between people)*, and *FinancialInstitution*.

In addition to the entities that OpenCalais detects, we implement a rule-based detection system, which performs matching based on regular expressions. In support of the use case presented earlier, we focus on detecting the names of terrorists and their membership with particular terrorist organizations. Figure 4 shows some sample extraction patterns that we look for in order to associate a particular person with a terrorist organization. Our regular expressions patterns are based on a list of known terrorist organizations. We allow for minor spelling differences in the names of the organizations and assume if the spelling is within a threshold of character differences then the names are a match.



```
//1.  Al-qaeda top commander Osama bin Laden

rule:(terroristorganization + "\s+(" + adjectives + "\s+)*" +
person_role + "\s+" + person);

//2.  Osama bin Laden, the top Al-qaeda commander

rule:(person + ",\s+(a|an|the)\s+(" + adjectives + "\s+)*" +
terroristorganization + "\s+" + person_role);
```

Figure 4 - Example Extraction Rules

Figure 4 shows two example rules. Before each rule, a sample phrase that matches against that rule is listed. In each rule, there are keywords that are used to match types of words in a phrase. For example, the keyword *terroristorganization* matches with a list of known terrorist organizations that is read from a configuration file. The keyword *adjectives* represents a list of known common adjectives that are used in the context of terrorist members and leaders. The *person* keyword refers to the name of a person. We use the detection of persons from OpenCalais. *Person_role* represents various roles or positions that a person can take (e.g. *member*, *leader*, *director*). Other standard regular expression symbols are used for matching whitespace, articles, and punctuation.

As the assertions are extracted, they are inserted into an intermediate model using Jena [18]. This Jena model maintains occurrence frequency for each assertion as well as the pertinent information for the assertion (e.g. terrorist organization name, person name, or communication status). Once the assertions are stored in an intermediate model, they are then transferred to the semantic model Web service for integration with other domains information and subsequent contextual analysis.

We ran across some limitations of the OpenCalais Web service. OpenCalais does perform co-reference resolution with pronouns, but has difficulty when the reference does not utilize a personal pronoun but instead using a plain noun to refer to an object. For example, a paragraph may refer to 'the president' several times without reference to the person's actual name. In this case, OpenCalais will have difficulty in making the connection with the particular reference to the president's actual name.

Another limitation of OpenCalais is that the relationships it extracts are limited to certain pre-defined relationship types that exist in the RDF schema. Although the schema is quite extensive, there are certain instances where it is more powerful to implement a domain-specific relationship extraction technique - like our rule-based technique. In our approach, we used a rule-based system to extract information regarding terrorists. This is quite a domain specific extraction and could not be extracted using OpenCalais.

NLP Web Service Implementation

The NLP Web service is implemented as a SOAP service that can be invoked by the end-user application, or perhaps another service. There are two primary components of the NLP service: a thin wrapper layer and the backend server module. This section describes both of these service modules.

The thin wrapper layer is a Web service that provides a Web interface to the user application. This layer provides an API which allows applications to perform various NLP related and file system related tasks. This layer communicates with the backend NLP server over a socket-layer protocol.

The backend NLP server module is the component of the NLP server which performs all of the primary work. It is multi-threaded to handle requests from multiple requests.

There are two primary service calls which the backend server handles: *BatchDownload* and *RunBatch*. *BatchDownload* is an API call which invokes the backend server to download a number of articles.

RunBatch calls the OpenCalais Web service for each of the documents that was downloaded using *BatchDownload*. OpenCalais returns an RDF file containing the extracted relationships. This RDF file is cached locally for future use and the assertions within a particular RDF file are compiled into a statistical list of assertions for a given batch.

SEMANTIC ENVIRONMENT

To extend the solution into the realm of semantic analysis, the NLP environment described above is partnered with the second component of this two-part approach, the semantic environment. This additional environment consists of a domain-specific OWL-based context model that is managed within a Jena-based platform equipped with an embedded reasoner. The resulting model and inference capability is collectively exposed as an invocable Web service.

The following section presents the representational paradigm selected to represent the respective domain ontology together with applicable inference logic. Within this section, several key modeling capabilities are highlighted and related to the information extraction problem at hand.

The OWL Representational Paradigm

At the heart of the semantic platform is the context model, or ontology, which contains the concepts necessary to describe and reason about the extracted information fragments produced by the enhanced NLP capability. Particular attention was given to the selection of the appropriate representational paradigm suitable to support the flexibility and inference involved in semantic analysis. This paradigm must provide a somewhat relaxed view of classification along with native support for embedding inference logic within the very concepts that it semantically relates to. Following investigation into several more traditional paradigms, we decided upon OWL as the representational language and execution platform suitable for this research.

OWL is a semantic markup language whose primary purpose is to facilitate the publishing and sharing of ontologies across the World Wide Web (WWW). OWL is intended to be used by software applications that need to process Web-based content in a meaningful manner. That is, OWL-based content is designed to be machine-interpretable.

A typical OWL environment consists of several key components that support both model development as well as subsequent execution. A characteristic OWL execution platform consists of a triple-store where model content is persisted, a reasoner capable of inferring additional concepts based on embedded and externalized logic, and a query engine used to ask questions regarding model content. Together, these components form a cohesive platform for the development and execution of semantic content.

Perhaps the most significant component of any OWL environment is the reasoner and as such warrants a more detailed description of its tasks. As the name implies, the main function of this component is to essentially reason about a given OWL model and its associated content. More specifically, an OWL reasoner processes class definitions, individuals, and rules in an effort to accomplish two primary objectives. The first of these tasks is to identify any logical inconsistencies existing within the model definition and its use. As the platform produced by this research supports user-driven importation of ontologies applicable to the target analysis domain (e.g., intelligence, logistics, command and control, etc.), this feature can be used to verify decidability and logical consistency of such models. The second task performed by the reasoner is to identify any additional knowledge that can be automatically inferred based on the logic accompanying the model definition in conjunction with associated content. This additional knowledge can include subsumption and association relationships along with the adjustment of classification of various scenario content. Although only beginning to emerge within the timeframe of this research, some reasoners are beginning to have the ability to not only infer additional content, but to retract previously inferred content whose truth can no longer be established (i.e., truth maintenance) [19]. This is a crucial feature for any practical application of automated reasoning as establishing what is no longer true is as important as the initial inference that produced it.

Above and beyond the components comprising its typical platform, the OWL representational paradigm supports several very powerful concepts that can be successfully exploited by the information extraction process. These concepts provide the flexibility to represent as of yet unclassifiable extractions as well as to repeatedly adjust the definitions of those that can be classified at the present time.

Multiple Classification

Multiple classification is the ability for something to be simultaneously defined under two or more classifications. This is a very powerful capability and has significant implications on the manner in which representational models are developed. Unlike traditional, more rigid modeling paradigms where inheritance must be employed as a means for the specialization of concepts, OWL modelers enjoy a much more flexible environment without concern for relating classifications in order to support a single entity exhibiting features defined across multiple classifications. Once qualification rules are embedded within class definitions, the management of exactly which classifications are appropriate at any given time can be conveniently offloaded onto the OWL reasoner.

Dynamic Classification

Dynamic classification is the ability for the classification of something to be adjusted throughout time. In contrast to the traditional approach of re-instantiating an entity under a new classification, dynamic classification offers the means to preserve referential integrity by maintaining the existence of the original entity by only changing which type characteristics are currently applicable. This capability goes hand-in-hand with multiple classification in creating a dynamic environment where extracted facts can effectively mutate throughout their lifecycle as additional knowledge is encountered. Like management of multiple classification, determining exactly what classification(s) an OWL object qualifies for at any point in time is typically the responsibility of the OWL reasoner.

Logical Assumptions

Traditional database systems operate under a set of assumptions in order to enable the query engine to return meaningful responses. These suppositions include the closed world assumption, the unique name assumption and the domain closure assumption.

The closed world assumption states that if a statement cannot be proven true, given the information in the database, then it must be false. The unique name assumption states that two distinct constants designate two different objects within the given universe. The domain closure assumption states that there are no other objects in the universe than those designated by constants within the database.

These assumptions were reasonable in a world where a database represented all the information available about a given domain and no external information sources were needed to perform the functions of any database application. Since this time, however, the Internet has become a major source of information with the effectiveness of many applications being based on access to external information from sources that may be unknown at the time of application design. This has required a different kind of knowledge representation, capable of dealing with the openness of the Internet. The open world assumption was adopted to allow for the relaxation of the constraints imposed by the closed world assumption. Along with the open world assumption, the other two assumptions were consequently also relaxed.

Under an open world assumption, all things are possible unless asserted otherwise. This is in stark contrast to traditional decision-support paradigms where the volume and extent of considerations is limited to what is explicitly asserted to be true about the world at any given time. Although operating under an open world assumption has implications on model development, it is primarily model usage and interpretation that is affected since logic operating across such a model must refrain from assuming too much and be receptive to the possibility that some information is yet to be discovered.

Unconstrained Composition of Features

The ability to describe the characteristics of an entity in a manner unbound by available definitions can be very helpful when dealing with evolving knowledge within an open environment. With environments equipped with such flexibility, information extractions can be characterized and continuously re-characterized in a manner unconstrained by current classification(s) or known concepts. Working in conjunction with dynamic classification, as an entity's characteristics evolve or otherwise change, so may its alignment with available definitions. As described earlier, determining exactly what changes in classification are appropriate is typically the responsibility of the reasoner.

The complementary partnership between this set of representational concepts is one of the most powerful aspects of the OWL paradigm. Collectively, these mechanisms support a progressive and fluid understanding of extracted information fragments within a dynamic and constantly evolving context.

Semantic Model Service

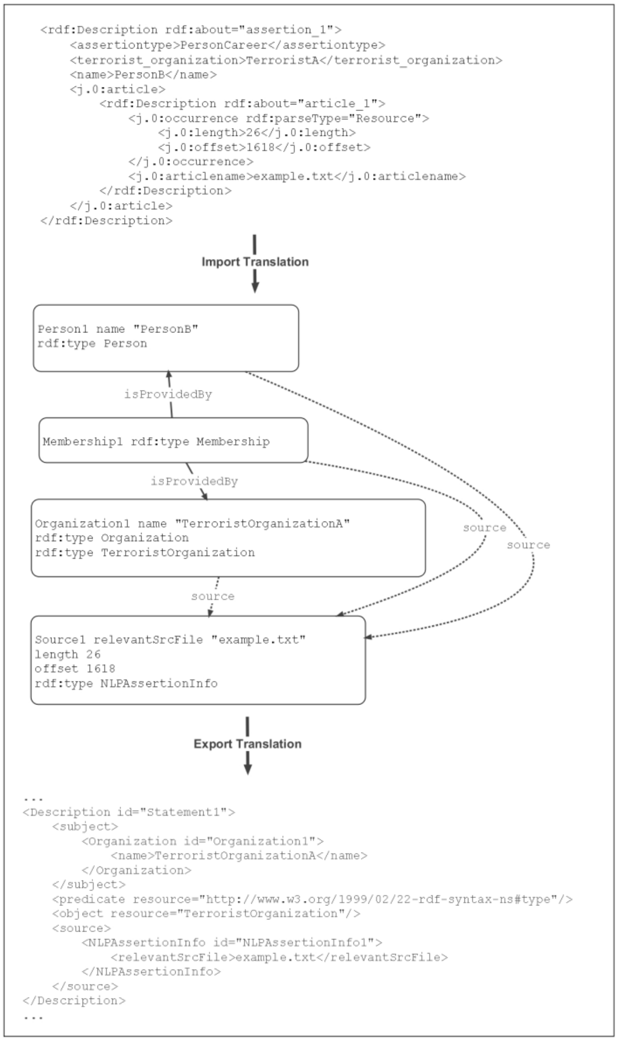
The Semantic Model Service contains the ontology and reasoning capabilities for our system. Its structure is a Jena RDF triple store embedded in a Web Service. This service takes information that has been extracted by the NLP component and unifies it with existing facts. Using multiple extractions from different sources allows us to create a more complete understanding of entities within the domain. This unified information can also be used as a basis for reasoning.

The Semantic Model Service has components to import the NLP extractions and translate them to the ontology of the current domain, uses SPARQL queries to export knowledge to the user interface, and manages the lifecycle of facts within the triple store. To enable reasoning over OWL concepts a domain specific OWL ontology is also imported.

Figure 5 illustrates the translation of knowledge through import and export processes. These translations are necessary because each component has different perspectives on the structure of the data. The NLP component views the data as discrete extractions and its terminology is based on the terminology of the tools it used for the extraction. The Semantic Model Service therefore needs to contain the knowledge to translate the extraction into the current domain. The structure of the export reflects a focused subset of the domain presented in XML and structured in an object oriented fashion.

Source Enhancements for Reasoning

The Semantic Model Service has the additional task of creating and maintaining *provenance information*. In our context provenance information is any additional data about a fact that describes the fact's context such as its source, reliably, or how it was created. We use RDF reification to attach provenance information to RDF triples. A survey for state of the art for the use of RDF reification for provenance is described in [19]. Work by [20] to increase the efficiency of reification shows the maturity of technologies that surround reification.



```
<rdf:Description rdf:about="assertion_1">
    <assertiontype>PersonCareer</assertiontype>
    <terrorist_organization>TerroristA</terrorist_organization>
    <name>PersonB</name>
    <j.0:article>
        <rdf:Description rdf:about="article_1">
            <j.0:occurrence rdf:parseType="Resource">
                <j.0:length>26</j.0:length>
                <j.0:offset>1618</j.0:offset>
            </j.0:occurrence>
            <j.0:articlename>example.txt</j.0:articlename>
        </rdf:Description>
    </j.0:article>
</rdf:Description>
```

```
Person1 name "PersonB"
rdf:type Person
```

```
Membership1 rdf:type Membership
```

```
Organization1 name "TerroristOrganizationA"
rdf:type Organization
rdf:type TerroristOrganization
```

```
Source1 relevantSrcFile "example.txt"
length 26
offset 1618
rdf:type NLPAssertionInfo
```

```
...
<Description id="Statement1">
    <subject>
        <Organization id="Organization1">
            <name>TerroristOrganizationA</name>
        </Organization>
    </subject>
    <predicate resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"/>
    <object resource="TerroristOrganization"/>
    <source>
        <NLPAssertionInfo id="NLPAssertionInfo1">
            <relevantSrcFile>example.txt</relevantSrcFile>
        </NLPAssertionInfo>
    </source>
</Description>
...
```

Figure 5 - Translation of Facts

The need for RDF reification is discussed in the papers cited above. An RDF triple represents a single fact. This bare fact stored as a triple cannot express information about its source, reliability, or how it was derived. Our system has the ability to make information available through a wide array of sources using NLP. Some sources may have incorrect information or be unreliable. Because of this in our system provenance information is as important as the fact it is tied to.

To express provenance each fact within our semantic repository is annotated with a source object. This annotation could be applied through the use of named triples, named graphs, or RDF reification. In our case we use RDF reification, since it can be used with any type of triple store. Unlike [19] the reification and provenance information are stored with the originating fact together in a single ontology.

Annotation of facts is just the first step to maintaining provenance in a semantic knowledge base. If we are to allow for reasoning over our semantic repository new facts that are inferred will need source data as well. A new fact's source is the inference itself. That inference as well as the facts used in the inference need to be made available to the user. Because of this we have taken the rules for typical OWL inference and extended them to create source information as well. The new source references the facts that were used in the inference as well as the source of the inference, such as a specific OWL axiom that the rule enforces. We can maintain provenance through extended inference rules because in our case provenance applies only to the assertion component (ABox) of our ontology. Figure 6 demonstrates an example of maintaining source information across inference.

This technique radically increases the size of rules for OWL inference. We have yet to discover its suitability for large datasets. It is possible that systems that deal with reification naively with named triples will allow for better scalability such as BigOWLIM [21]. Future work will need to examine these tradeoffs. Storing provenance alongside its related fact can be expensive but can also have its advantages. For example it is easy to create rule sets that take provenance into account when making inferences over the fact base. This could be important for agents that need to take reliability information into account before making recommendations or inserting new facts.

THE ROLE OF HUMAN INTERACTION

The basic design principle of this research relies on the collaboration between the human user and a set of tools, which can be configured at run-time, depending on user needs, to perform different tasks. The accuracy levels of existing NLP tools make user involvement essential in providing meaning and guiding both the extraction and the

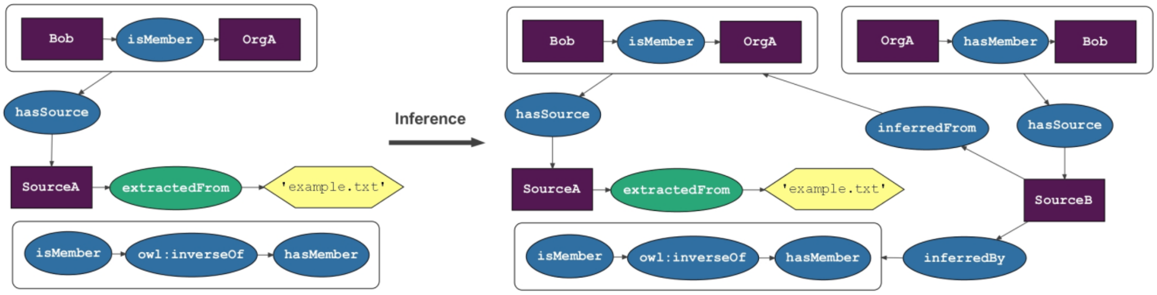


Figure 6 - Example Inference

inference processes. The application is developed as a Web application for availability to multiple users and easy access. Another design objective is to provide continuous visibility for the user into the process of extraction and analysis. The user can interact with the system at any time to guide the extraction process, change the identification of some objects, merge redundant information, or create new associations among identified objects. The user can also guide the inference process by providing additional domain knowledge, making manual connections, or modifying automatically created ones.

The analysis results are presented in tables as well as in a graph, depicting the extracted concepts and their relationships. The inferred objects and relationships are included in the presentation and marked as ‘inferred’.

In addition to the NLP extraction and the inference engine, the application offers tools to assist the user in the analysis, by finding out more information about specific concepts or identifying the types of some concepts that may have been missed by the extraction process.

One tool determines whether two or more words appear together as a single term. This tool is based on Google search. For example, to find out whether the ‘gateway of India’ is a landmark or just an expression, the tool performs a search on Google using the keywords: gateway of India, then examines the first two hundred results to find out how often the words appear in the given sequence.

Another tool is based on Wikipedia search, to find out whether an extracted concept is a place, organization, person, or another type that exists in the ontology. This tool takes advantage of the structure of a Wikipedia page, which provides basic information in a structured format for a large collection of concepts.

The ability for the user to configure certain aspects of the system is an integral part of the user experience in a collaborative intelligent system. The user experience in this application includes the ability to configure some aspects of the system to suit the user needs for a given task. On the NLP extraction side, the user can control the NLP server and stop it and start it at any time during execution. The user can also control the batch jobs that are running on the server and stop any job or delete it. In addition, the user can view the extraction rules and edit them to introduce new rules. The internal lists, which are used by the system to determine the specific types of organizations, can also be edited. These lists assist the extraction server in determining terrorist organizations, financial institutions and other types of organizations.

FUTURE WORK

The current platform performs NLP extraction and inference in a specific domain, which is intelligence analysis. The objective for the platform is to be domain-independent and flexible to meet the requirements across a variety of domains. To achieve the goal of domain-independence, the platform must allow the addition of onotologies at runtime with the ability to edit ontologies within a session to adapt to the analysis requirements of the given domain. The platform must also support editing the extracted entities to merge instances that refer to the same entity, e.g. different spellings of the same name, or to adjust the identification of an entity. Apart from these capabilities, it is anticipated that the next version of this platform will support the notion of a confidence level for the extracted entities, based on the source and the appearance of the same entity in multiple sources. Other factors, such as connections to well-known entities may also contribute to this rating as well.

In terms of reasoning capabilities, two new features will be added to the platform: user interaction with agents and truth maintenance. Agents perform their analysis in an autonomous manner, but there is a need for the user to ask specific questions of agents or direct their analysis by providing priorities or additional information. Truth maintenance keeps track of the dependencies among information items and adjusts them as new information is asserted or retracted to maintain the integrity of the current state of the ontology.

In addition, the platform will be deployed to a cloud computing environment making the individual services available to external applications as well as empowering them with additional resources. Both the NLP extraction service and the inference service are generic in form and can be used either separately or combined to build other applications. It is also expected that the cloud computing environment will make available additional capabilities that can enhance the overall utility and performance of the platform. These services include security, single sign-on (SSO), intelligent routing, load balancing and location transparency among other services.

CONCLUSION

Traditional NLP offers mechanisms to extract information from a collection of documents with the ability to recognize basic types of entities. However, in order to perform intelligent analysis on these essentially discrete extractions, a context must be provided along with some guidelines for the analysis. We partner existing NLP capabilities with an environment suitable for intelligent analysis that is configurable with a user-provided OWL ontology and associated inference logic. Deployed as an SOA set of capabilities, the resulting hybrid environment provides users with the ability to extract and subsequently analyze meaningful information from natural language content obtained from a potentially diverse set of sources (e.g. news articles, intelligence reports, blogs). Throughout the entire extraction and analysis process, users are provided extensive visibility and governance over decisions and conclusions derived within the system. The resulting environment forms a collaborative partnership where human analysts receive vital assistance in extracting meaningful information from what amounts to a sea of natural language content.

REFERENCES

[1] Wikileaks Afghan War Diary (2010). Wikileaks Organization, http://wikileaks.org/wiki/Afghan_War_Diary,_2004-2010; accessed Aug. 2010

[2] Schmitt, E. U.S. Tells WikiLeaks to Return Afghan War Logs. New York Times, http://www.nytimes.com/2010/08/06/world/asia/06wiki.html August 5, 2010,

[3] LingPipe. Alias-i, http://alias-i.com/lingpipe/, visited 8/10/2010.

[4] OpenNLP. http://opennlp.sourceforge.net/, Visited 8/10/2010.

[5] GATE. University of Sheffield, Computer Science Department. http://gate.ac.uk, visited 8/10/2010.

[6] Apache UIMA. Apache Software Foundation. http://uima.apache.org, visited 8/10/2010.

[7] ClearTK. Center for Computational Language and Education Research (CLEAR), University of Colorado, Boulder. http://code.google.com/p/cleartk/, visited 8/10/2010.

[8] Wimalasuriya, Daya C. and Dou, Dejing. Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science, June 2010, 36: 306-323.

[9] Lin, T., Etzioni, O., and Fogarty, J., Identifying interesting assertions from the web. In CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management, pages 1787–1790, New York, NY, USA. ACM, 2009.

[10] Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[11] Cuenca Grau, B., OWL 1.1 Web Ontology Language Tractable Fragments. http://www.webont.org/owl/1.1/tractable.html#7, 2007.

[12] OpenCalais. Thomson Reuters. http://www.opencalais.com, Visited 8/10/2010.

[13] OWL Specification. W3C, http://www.w3.org/TR/OWL2-overview/,October 2009.

[14] RDF/XML Syntax specification. http://www.w3.org/TR/REC-rdf-syntax/, February 2004.

[15] NewsExplorer. Europe Media Monitor, http://emm.newsexplorer.eu/, visited 8/10/2010.

[16] AlchemyAPI. Orchestr8, http://www.alchemyapi.com/, visited 8/10/2010.

[17] OpenCalais RDF. Thomson Reuters. http://s.opencalais.com/1/pred/, Visited 8/10/2010.

[18] Jena. Hewlett-Packard Development Company, LP, http://jena.sourceforge.net/, visited 8/10/2010.

[19] Vacura M. and Svatek V., Ontology Based Tracking and Propagation of Provenance NDT 2010, Part I, CCIS 87, pp. 489–496, 2010.

[20] Ramanujam S., Gupta A., Khan L., Seida S., Thuraisingham B., Relationalization of provenance data in complex RDF Reification Nodes, August, 2010

[21] BigOWLIM. Ontotext AD, http://www.ontotext.com/owlim/big/, visited 8/10/2010.

BIOGRAPHIES

Dr. Hisham Assal *received his Ph.D. in Computer-Aided Design from the University of California, Los Angeles (UCLA) in 1996. As a Senior Software Engineer in the Collaborative Agent Design Research Center at Cal Poly, Dr. Assal designs and develops distributed, collaborative decision-support systems for military and industry applications. His interests include knowledge representation, knowledge management, intelligent agents, and complex systems.*

Franz J. Kurfess *joined the Computer Science Department of California Polytechnic State University in the summer of 2000, after a short stay with Concordia University in Montreal, Canada, and a longer stay with the New Jersey Institute of Technology. Before that, he spent some time with the University of Ulm, Germany, the International Computer Science Institute in Berkeley, CA, and the Technical University in Munich, where he obtained his M.S. and Ph.D. in Computer Science. His main areas of research are Artificial Intelligence and Human-Computer Interaction, with particular interest in the usability and interaction aspects of knowledge-intensive systems.*

John Seng *is an associate professor in the computer science department at Cal Poly State University, San Luis Obispo. His research interests include natural language processing and machine vision.*

Emily Schwarz *is a Master's student at California State Polytechnic University pursuing a degree in computer science. She received her Bachelor of Science degree in computer science from San Diego State University in 2007. Her interests include artificial intelligence and human-computer interaction.*

Kym J. Pohl *is a principal software engineer with Tapestry Solutions, a subsidiary of Boeing. His current focus is agent-based, collaborative decision-support systems with particular interest in representation and collaborative architectures. Following an undergraduate degree in Computer Science he earned Master's degrees in both Computer Science and Architectural Design. Over the past two decades he has provided technical leadership and consultation in the design and development of a number of context-oriented, decision-support systems for the US Department of Defense and industry. Additionally, Kym is periodically involved in collaborative research efforts with members of academia as well as industry.*