

Semantic Representation and Visualization of Natural Language Text

Jason Wells
Semantic Research, Inc.

Abstract

This paper describes at a high level a research investigation into the integration of several GOTS, COTS and open source natural language processing technologies with a system offering novel forms of semantic network representation and visualization. The integrated system described herein, called Semantica® Enterprise, provides analysts the capability to extract arbitrary natural language text for import into a semantic network via an automated, user-supervised approach. Analysts can process structured, semi-structured and unstructured text sources and fuse the extracted information together into a unified representation according to user-defined ontologies. Networks are visualized on screen in various ways, allowing for the discovery and establishment of relationships within newly extracted information. A semantic search and indexing capability supports information retrieval from very large repositories of structured and unstructured content. The integrated system provides analysts with new tools and techniques for exploiting information within documents and extracted semantic content in support of an investigation.

Introduction

One of the long-standing objectives of the intelligence community has been to develop an operational capability allowing analysts to leverage information obtained from natural language text via natural language processing (NLP) in a highly visual, semantically rich computing environment. More specifically, there is a need for an integrated system that offers:

- The ability to represent extracted natural language text semantically
- The ability to display and manipulate extracted information visually
- Indexing and search capabilities for all semantic information, both as original natural language text and extractions
- Support for named entity recognition, machine translation and transliteration from multiple best-in-class technology providers including, but not limited to, GOTS NLP platforms
- The ability to apply these services uniformly to unstructured, semi-structured and structured data, supporting data fusion from various sources
- A web-based user interface, supporting a fully unified user experience making these integrated capabilities easily accessible from a browser

In support of these requirements, our technical approach has been to extend the existing technology of Semantica Enterprise where needed. Prior to this investigation, this system had no built-in support for natural language processing, so the primary tasks involved integrating several existing NLP services and components into the core

system. This paper describes how the integrated system functions and how it may be used to support the analyst's operational workflow.

Research Agenda

The overarching objective of this research is to empower analysts as they conduct investigations. While sophisticated NLP capabilities have been generally available for some time, the ability to leverage them calls for their effective integration into a general-purpose analytical platform. For this investigation, we explored the value of combining NLP and its informational product, extracted entities, with semantic network visualization in a scalable, web-based environment. Our research agenda primarily addresses the following questions:

1. How can natural language documents and their entities be represented and persisted as semantic networks?
2. How can this structured and unstructured information be fused and visualized?
3. How can combining these technologies provide new capabilities that help analysts?

Operational Workflow

In its most basic form, the analyst workflow begins with one or more natural language documents containing information that may be useful to the investigation. The documents are uploaded to the system, each becoming a node in the analyst's semantic network.

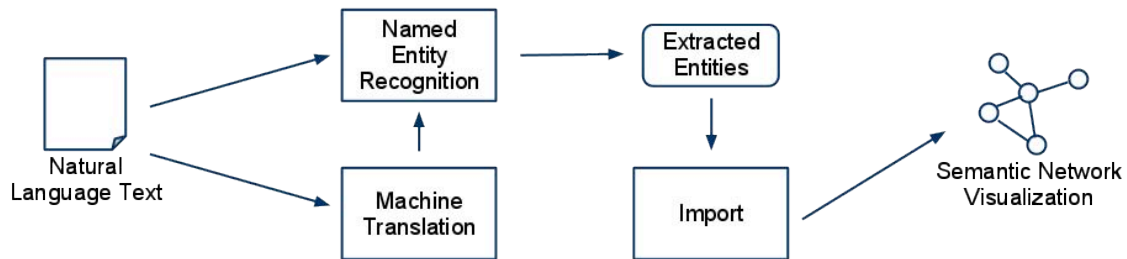


Figure 1. High level workflow.

If the document is in English, the analyst performs named entity recognition to extract entities from the document. If the document is in another language, the analyst performs machine translation to translate it to the target language, English, with extraction then performed implicitly on the translation. In both cases an extraction is produced which indicates the extracted entities within the original text.

The analyst is offered the option to import some or all of these entities into the semantic network. Once imported, the entities become new nodes in the network. At this point the information sourced from the provided documents is represented in semantic form, allowing it to be fused directly with information from other sources and displayed using the various graphical visualizations described below.

Semantic Network Data Model

To represent information internally, the system implements a data model based on the notion of the semantic network (1). This data model is central to the system's utility as an information analysis platform. It serves as the flexible and scalable foundation for a suite of visualizations essential to the analytic process. But beyond its appeal as a means to graphical presentation of factual information, one of the fundamental rationales for the semantic network model is its suitability as a vehicle for seamless data fusion. As this model primarily specifies highly generalized syntactic formalisms, it supports a common, unified representation of information obtained from both structured and unstructured sources and from a diversity of schemas and vocabularies. This unites interconnected information, making it amenable to visualization, information retrieval and in-depth analysis using a common set of tools.

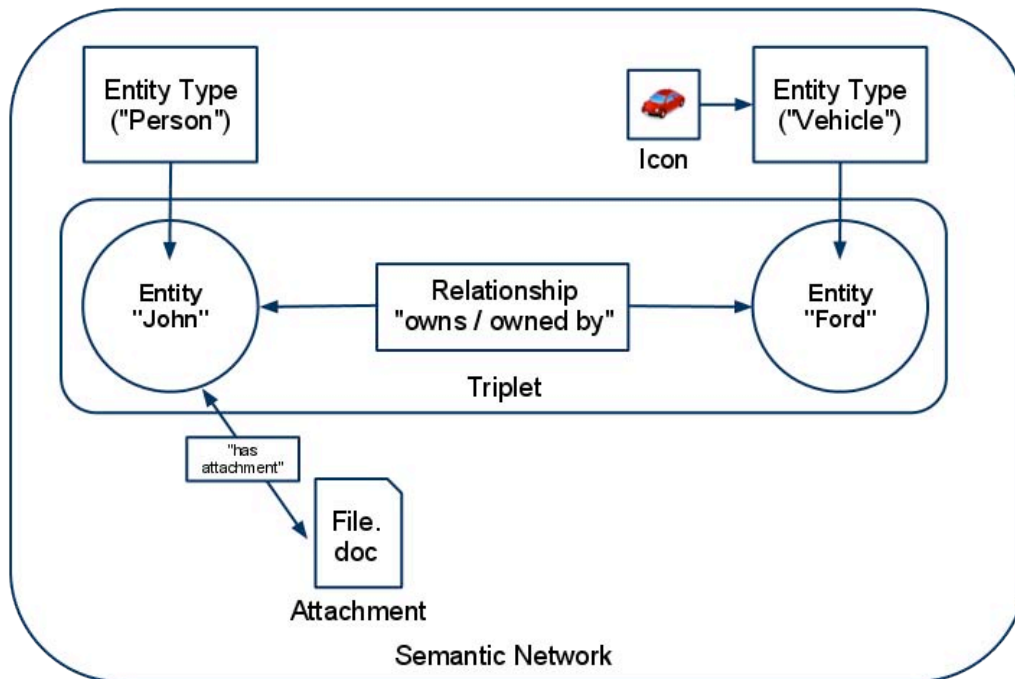


Figure 2. Simplified rendition of the semantic network data model.

In the semantic network model, extracted entities and relationships are maintained structurally as an undirected graph. A full description of the Semantica Enterprise data model is beyond the scope of this paper, but the most relevant primitive elements of a semantic network include:

Entity

The entity is the fundamental iota of meaning in the system. Serving as the nodes of the semantic network, entities may be created from imported or ingested data, NLP extraction or via manual data entry. Typically representing people, places, things, events or ideas, they are concrete instances of entity types defined in the network's ontology, described below. Entities can be members of multiple networks, allowing

the same facts to be referenced and leveraged in separate topical domains and investigations simultaneously.

Attachment

An attachment represents a file embedded in the semantic network. Like an entity, it exists as a node in the graph, allowing files to participate directly in the model as units of information without requiring prior extraction or conversion. Attachments may be joined in arbitrary relationships with other entities or attachments, and they appear much like entities across the user interface and in all semantic visualizations. Attachments may be created manually or ingested in bulk. They are subjected to NLP to extract entities for direct use in the network. Like entities, attachments can exist in multiple networks at once.

Triplet

Entity and attachment nodes are connected with explicitly named bidirectional relationships. When a specific relationship is established between two nodes, it forms a triplet, serving as an edge linking two particular entities. Entities may be members of any number of triplets. The process of joining them together with relationships creates the semantic content of the network. Like entities and attachments, triplets may exist in multiple networks.

Ontology

The content within the semantic network is expressed in terms of an ontology, its hierarchical system of types. It includes defined entity types and relationships used to categorize entities and triplets respectively. The system includes a general-purpose ontology and supports the creation of arbitrary user-defined domain ontologies. Any number of discrete semantic networks can share the same ontology. It may be modified at runtime from the application without causing data destruction or otherwise negatively affecting system behavior.

Integrated NLP Capabilities

For this investigation, the system concentrated on integrating two forms of NLP: named entity recognition (NER) and machine translation (MT). Both functions are made available to the analyst in the user interface and are invoked on demand for selected documents. A secondary transliteration capability is also made available as a fallback to incomplete or inaccurate translation (see “User-Supervised Approach” below).

Named Entity Recognition

NER is the process of identifying entities—persons, locations, etc.—in natural language text and assigning a type to each. This extracted information maps directly to the formalisms of the system’s semantic data model. The extracted entities can then be brought into the network as new semantic entities, taking types defined in the network’s ontology. These entities become nodes within the network visualizations, available for further analysis.

To integrate NER functionality, the system uses the embedded form of General Architecture for Text Engineering, or GATE Embedded (2). This open source component, which is integrated into the system internally, is widely used and provides a broad suite of language processing capabilities (3). It identifies and types entities in English text and returns the text annotated accordingly. If the type in question isn't in the ontology, it may be added automatically at import. GATE is highly effective at NER, consistently producing high scores for average precision, recall and F-measure in a variety of tests (4).

Machine Translation

MT is performed by two services: Language Now and Google Translate. Unlike NER, these services are external to the system application stack and are invoked remotely. Language Now supports automatic detection and translation of up to 50 languages, depending upon configuration (5). Google Translate supports a comparable 57 languages (6). This investigation evaluated translation from three source languages, well supported by both services: Spanish, Arabic and Chinese. For both services and all source languages, the target language was English.

Extracted Entity Import

After NER and/or MT is performed on the natural language text of the selected attachment, the results appear in the application as graphically annotated text with a side pane indicating the types and counts of extracted entities (see Figures 4, 5 and 6). Each entity's type is indicated in the resulting text by its color. This interface allows the analyst to review the extraction before any of the entities are added to the semantic network. The analyst selects the relevant extracted content and imports it into the network. The imported entities are then persisted as entity nodes in the network and become available for search retrieval, graphical visualization and data fusion. Multiple extractions may exist in the interface side by side, allowing the analyst to work with multiple documents simultaneously.

User-Supervised Approach

Natural language processing is well known to be an inexact process, and one of the issues we encountered was the challenge of accurately representing information extracted from unstructured text. In the majority of cases, entities in the document are identified and typed as the analyst expects. However, depending upon the nature and quality of the source content, the desired entities may not be extracted, undesired ones may be, and entities may not be typed as needed or in accordance with the network's ontology. The syntax, semantics and overall quality of translations can vary significantly across NLP services and by language.

In order to guide content effectively from the source text to the semantic network, a few simple corrective functions are provided in the application. The system can be engineered to support a more automated workflow, but allowing for adjustments as part of the normal workflow makes it possible for the analyst to resolve discrepancies and to realize more value from the integrated NLP functionality.

Manual Entity Extraction

In cases where the NLP functionality fails to identify an entity, the user interface offers the analyst the ability to handcraft an entity manually prior to import. The analyst selects the desired text and clicks a button to turn it into an entity. In the simplest case, this creates a new entity given the type “Unknown.” The option exists to create a new entity for a given type as well.

There are cases where the extraction produced the desired entity partially, perhaps due to encountering hyphens, spaces or other non-alphabetical characters. To correct this, the analyst selects the entity along with the adjacent unextracted text and clicks a button, producing a new entity that including all selected text. The typing for the original entity is maintained for the new entity.

Manual Entity Merge

Sometimes what is desired as a single entity is broken out as multiple entities by the extraction. To fix this, the analyst has the ability to select two or more extracted entities and merge them into a single entity.

Ontology Mapping

The ontology of the semantic network defines a set of entity types that describe extracted entities. Extracted entities may or may not correspond to these predefined types. Automatic type matching is based on entity type name. The application distinguishes matching and nonmatching entity types in the extraction results and offers the analyst several ways to map the extracted types to the types in the network’s ontology:

- An extracted entity type may be replaced by one defined in the ontology. To support this, the interface allows for dragging the type from the ontology onto the extracted type. Entities of the original type then import with the type from the ontology.
- An entity type in the ontology may be added to the extraction by dragging it onto the list of extracted types. The analyst may then reassign entities to this type by drag and drop.
- A new entity type may be created in the extraction results with entities given this type as described above. This causes the new type to be added to the ontology at import.

Manual Entity Typing

The analyst can drag entities from one type to another to change their type. The application also allows the analyst to create new entity types that were not produced by the original extraction.

Single vs. Multiple Entity Import

By default, importing takes all extracted entities and adds them to the network. Often, however, the analyst may wish to import only a subset of these. There are several options for indicating which of the entities should be imported that may be used separately or in combination:

- Drag a single entity from the extraction into the list of entities in the network. This action imports entities individually.
- Deselect particular entities or all entities of particular types.
- Deselect all entities and manually select particular entities or all entities of particular types.

Transliteration

If the translation service fails to translate a term, that term is left untranslated in the extraction display. The term may be in a non-Latin character set, perhaps unreadable to the analyst. Even so, the text may still be of use. As a fallback to translation, the system uses transliteration to convert the text to equivalent English phonemes. For example, if the Arabic word فنقط is provided to the Language Now service, it returns the transliteration *nqtah* along with the original Arabic text. The analyst may then indicate in the interface that the transliteration be imported as if it were a translated entity to incorporate it into the semantic network. This allows the entity to participate usefully in relationships with other entities without depending on a successful translation.

Source Content

The original source for all extracted natural language content is attachments. The capacity to perform entity extraction, translation and transliteration exists for unstructured, semi-structured and structured file data. It is also possible to extract from files contained in compressed file archives as well as metadata from image files.

These attachments take two forms. They may represent file data stored locally within the Semantica Enterprise cloud or may reference external Web content by URL. Regardless of the format of the source, extractions are fused seamlessly into the common underlying semantic representation.

Unstructured Data

The typical source content for NLP is unstructured natural language text contained in a written document. Many common document file formats are supported: Adobe PDF (.pdf), Microsoft Word (.doc and .docx), OpenDocument Text (.odt), RTF and plain text. In the case of binary file formats, the plain text is first read from the file and provided in that form to the NLP services.

Semi-Structured Data

In addition to unstructured natural language text, semi-structured content is amenable to NLP. The following semi-structured file formats are supported: HTML, XML, SVG, Microsoft Outlook (.msg) and Unix mailbox (.mbox).

Structured Data

Entities may also be extracted from structured content. For example, the following sample Spanish-language CSV file is uploaded as a file attachment:

la Policia	contrabandistas	maletero del coche	marihuana	Ensenada
Autoridades	los contrabandistas del drogas	kilogramo	cocaína	Puebla
Funcionarios	ladrones	tubo	heroína	Mexicali
el Ejercito	detenido	100 pastillas	oxicodona	Ciudad de Mexico
las Fuerzas Armadas	prisionero	comprimidos	drogas	Ecatepec de Morelos
jefe de la Policia	acusado	comprimidos	medicamento	León

Figure 3. Sample CSV data.

When configured to use Language Now, the system produces the following translation:

The police,smugglers,car trunk,**marijuana**,**Ensenada** Authorities,the smugglers of drugs,kilogram,**cocaine**,**Puebla** **Officials**,thieves,tube,**heroin**,**Mexicali** the **Army**,detenido,100 pills,**Oxycodone**,**Mexico City** the **Armed** Forces,prisionero,tablets,drugs,**Ecatepec** of **Morelos** **Police** **chief**,**accused**,tablets,medicine,**Lion**

- Drugs (3)
- City (4)
- Jobtitle (2)
- Organization (2)
- Unknown (5)
- Actions (1)

Figure 4. CSV translation (Language Now).

When configured to use Google Translate, the system produces this extraction:

the **Police**, smugglers, car trunk, **marijuana**, **Ensenada** Authorities, drug smugglers, KG, **cocaine**, **Puebla** officials, thieves, tube, heroine, **Mexicali** the **Army**, **arrested**, 100 pills, **Oxycodone**, **City** **Armed** Forces of **Mexico**, prisoner, tablets, drugs, **Ecatepec** de **Morelos** police **chief**, **accused**, tablets, medicine, **Len**

- Organization (2)
- Drugs (2)
- City (4)
- Actions (2)
- Unknown (4)
- Country (1)
- Jobtitle (1)
- Male (1)

Figure 5. CSV translation (Google Translate).

In addition to CSV, the following structured file formats are supported: tab-separated values (.tsv), Microsoft Excel spreadsheet (.xls and .xlsx) and OpenDocument spreadsheet (.ods).

Image Metadata

NLP on several common image file formats is supported, allowing NER, MT and transliteration to be performed for image metadata or embedded text. This metadata

may include geotagging, timestamps and other descriptive information embedded within the source image.

```
tiff:Make = Hipstamatic Software = John S Lens, Ina's 1969 Film, No Flash Model = 200
Date/Time Original = 2011:02:13 12:38:52 Scene Capture Type = Standard Exif Image
Width = 600 pixels Exif Version = 2.21 Component 1 = Y component: Quantization table
0, Sampling factors 2 horiz/2 vert Component 2 = Cb component: Quantization table 1,
Sampling factors 1 horiz/1 vert tiff:ImageLength = 600 GPS Latitude = 32°49'27.000046
Component 3 = Cr component: Quantization table 1, Sampling factors 1 horiz/1 vert
exif:Flash = false GPS Latitude Ref = N tiff:ResolutionUnit = Inch X Resolution = 72 dots
per inch Flash = Flash did not fire Date/Time Digitized = 2011:02:13 12:38:52
tiff:ImageWidth = 600 tiff:XResolution = 72.0 Image Width = 600 pixels Resolution Unit =
Inch GPS Longitude = 117°10'38.999977 GPS Longitude Ref = W tiff:Software = John S
Lens, Ina's 1969 Film, No Flash Number of Components = 3 Orientation = Top, left side
(Horizontal / normal) tiff:Model = 200 Color Space = sRGB Image Height = 600 pixels Data
Precision = 8 bits tiff:BitsPerSample = 8 geo:lat = 32.82417 tiff:YResolution = 72.0 YCbCr
Positioning = Center of pixel array Components Configuration = YCbCr
exif:DateTimeOriginal = 2011-02-13T12:38:52 Compression = JPEG (old-style) FlashPix
Version = 1.00 Thumbnail Offset = 564 bytes Exif Image Height = 600 pixels geo:long =
-117.1775 Thumbnail Length = 14947 bytes Thumbnail Data = [14947 bytes of thumbnail
data] Content-Type = image/jpeg Y Resolution = 72 dots per inch tiff:Orientation = 1
```

Figure 6. JPEG image metadata extraction.

These graphical formats are supported: JPEG, TIFF, PNG, BMP, Microsoft PowerPoint (.ppt and .pptx), OpenDocument Presentation (.odp) and OpenDocument Graphics (.odg).

URLs and Archives

NLP functions may be applied to content hosted externally on the Web. URL attachments are used to extract entities directly from external Web resources. For URL attachments, the data is first fetched and then processed identically to locally persisted data. Unextracted and unimported data is discarded. The URL may refer to content stored in any of the supported formats above.

A compressed ZIP archive of multiple files may exist in the network as a single attachment. The system supports NLP against ZIP archives, allowing the analyst to treat the archive as a single file for NLP purposes. The system automatically decompresses the files within the ZIP archive prior to submission to NLP services. Any unsupported file formats that may be included are ignored. The combined extractions are presented to the analyst as a single extraction.

Data Fusion

One of the longstanding problems in analysis is that information about a particular topic that belongs together may only exist in various separate and incompatible forms. For example, information about a particular person may exist in both a relational database table and a text document, but there has been no general-purpose way to relate information from one to the other directly; there is no SQL query to join

a table with a paragraph of natural language text. Because of this information-independent stovepiping, relationships between facts—often crucial ones—may never be established, discovered or even looked for. Even within the same basic structure, information may be schematically stovepiped. For example, two instances of the same relational database can represent the notion of a person differently according to their respective schemas; perhaps one schema uses a single denormalized table but the other uses three. Historically, there has been no general solution to this problem, so one-off schema-specific mappings have been required to unite information across the divide. These problems severely hinder analysis and are the motivation for the format- and schema-independent data fusion capability provided by the system.

A simple example of data fusion can be found in Figures 9 and 10 below, in which entities extracted from natural language documents are fused with latitude/longitude coordinates obtained via a separate GIS lookup. Despite coming from different kinds of sources, the totality of information is represented in a single form, as typed entities participating in triplets.

Information from data sources as varied as relational databases, XML documents, message traffic and unstructured text can be mapped to the same underlying semantic representation. Support for multiple runtime-modifiable ontologies allows information structured in accordance with incompatible schemas to coexist in the same model. In this way the semantic model serves as a *lingua franca* between the various sources, allowing for meaningful data fusion across a broad spectrum of formats.

Web-Based Semantic Network Visualization

Clear and powerful visualization is key to effective analysis of large amounts of complex and potentially highly interrelated information. The system provides a collection of graphical views to discover and analyze extracted entities and the relationships between them visually.

The analyst interacts with the system through a web application accessible from a standard browser. All integrated functionality – extraction, translation, and visual display – is presented in a unified, web-based user interface. This interface is designed such that basic system functionality can be easily learned and applied by technically unskilled analysts. A separate web application is used to configure, monitor and administer the system.

The semantic network visualizations are integrated natively into the main interface, each appearing as a tabbed pane in a multi-tab presentation. This investigation explored five forms of semantic network visualization used to analyze the information extracted from the source documents. Each is described below.

Entity View

One of the main difficulties in visualizing a semantic network is information overload. Semantic networks can be densely interconnected, appearing as undifferentiated “hairballs” in naïve graph displays. Another complication is that any two entities may

be connected, regardless of position in the graph, producing network topologies with no natural mapping to 2D or 3D spaces. Thus attempting to draw even a simple network on a 2D surface, such as an ordinary computer display, usually results in a nonplanar graph. Relationships between entities are drawn crossing and overlapping each other, producing a cluttered and potentially unusable view.

To avoid these pitfalls, Semantica Enterprise provides a visualization known as Entity View, shown in the figure below.

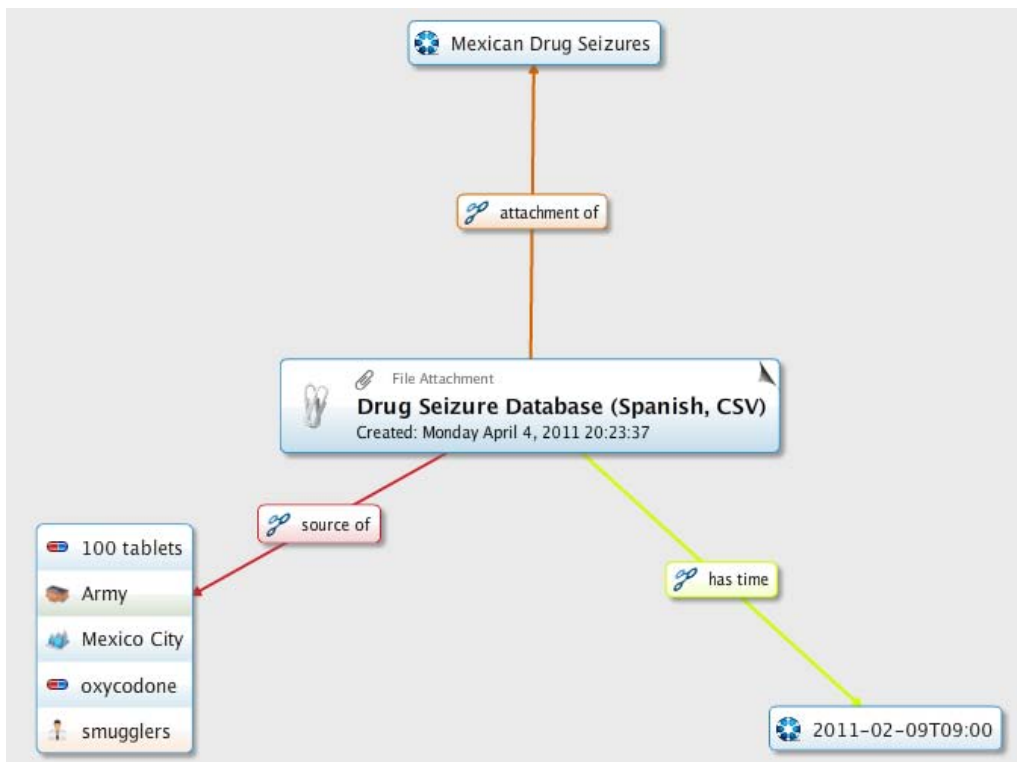


Figure 7. Entity View visualization of a simple MT/NER extraction.

Entity View draws a planar subgraph of the network, providing a single central entity and arrays all entities directly linked to it in a radial fashion around it. This makes a small part of the overall network amenable to 2D visualization. In this way the view focuses attention directly on a single entity while retaining awareness of its adjacent network context. Typing information is conveyed by the associated icon and color. Related entities sharing a common relationship with the center entity are grouped together to reduce visual clutter, as with those related by the relationship “source of” in Figure 7. Clicking on a related entity causes it to become the center entity, displacing the previous center entity. In this manner the analyst may use this view to traverse the network from one entity to the next.

Multiple instances of Entity View may be open in the application simultaneously. The subject of the view may be any entity of interest within the network. This view may also be used to author and modify network content by hand, establish relationships between extracted entities, between extracted entities and preexisting content in the semantic network or between entities and the source document itself.

Card View

Like Entity View, Card View takes a single entity or attachment as its subject. It presents the entity in the form of a detailed report, in which the content of the report is provided by the content of its immediate (one degree) vicinity. It displays essentially the same information found in Entity View: intrinsic information, such as the entity's name, type, synonyms and properties; and extrinsic information, such as tags and relationships to other entities. Triplets are grouped by relationship, similar to the radial groupings in Entity View.

File Attachment

Drug Seizure Database (Spanish, CSV)

Mexican Drug Seizures DB

urgent

▶ Attachments

▼ Triplets

related

▼ attachment of

Name	Type
Mexican Drug Seizures	Entity

▼ has time

Name	Type
2011-02-09T09:00	Time

▼ source of

Name	Type
100 tablets	Drug
Army	Military Organization
Mexico City	City
oxycodone	Drug
smugglers	Person

Figure 8. Card View of an attachment with relationships to other entities.

Each entity in a network may have its own Card View. Multiple instances of this view may be open in the interface simultaneously. The analyst can use this view to edit the entity manually and to establish or change relationships between entities.

Network View

The Network View visualization builds upon the notion of a concept map: a diagram displaying an arbitrary collection of entities and their relationships as a graph. It is the traditional and perhaps the most natural visual presentation of a semantic network. It can be used to display upwards of thousands of entities and relationships as a cohesive whole.

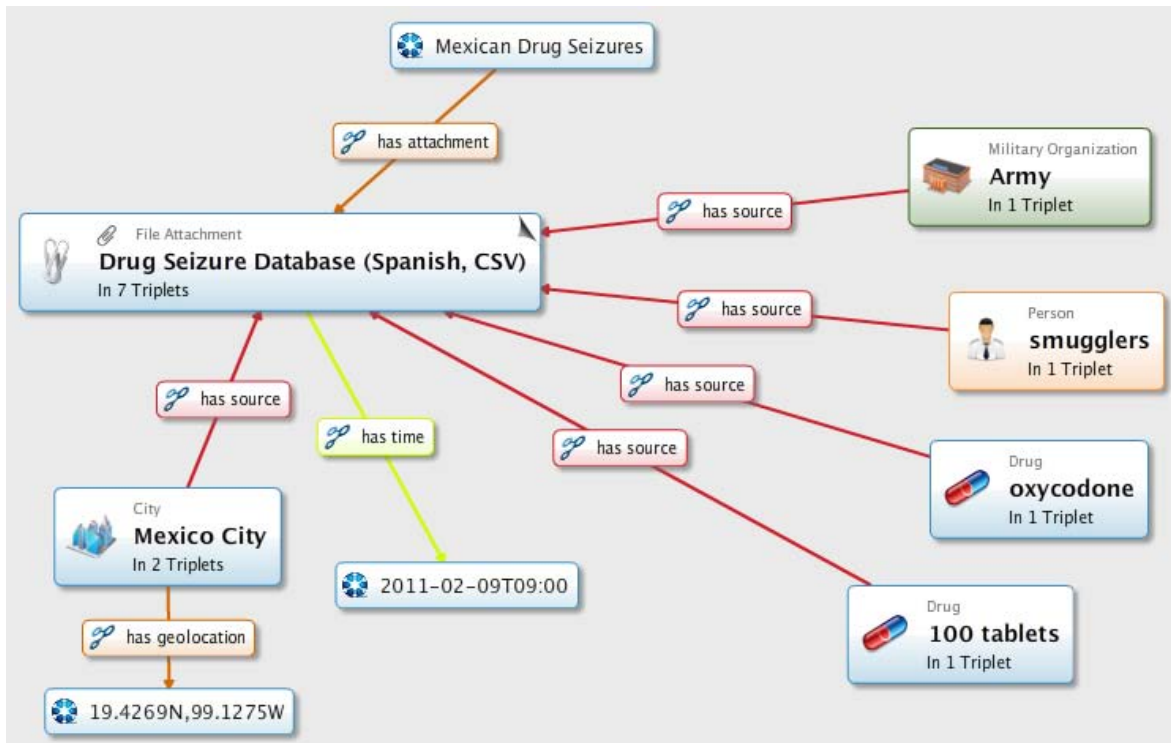


Figure 9. Network View of extracted entities and their source document.

While Entity View and Card View focus on the individual entity, the subject of Network View is the full breadth and depth of information represented in the underlying network. It offers a different tradeoff than Entity View, exchanging simplicity for completeness.

In Network View, networks of virtually any size might be nonplanar and visually busy or otherwise complex, necessitating sophisticated layout options. To address this, the view provides four layout algorithms that the analyst may use directly to organize the displayed information as desired. A force-directed layout is employed by default, seeking to render the graph with as few crossing edges as possible. This layout is often capable of producing an uncluttered presentation, but its effectiveness is highly dependent upon the topology of the graph. A grid layout orders the entities as an alphabetically sorted table. A circle layout arranges entities in a circle with relationships connecting within. There is also a hierarchical layout, displaying entities and relationships as a horizontal tree rooted from a selected entity.

The four layouts may be employed globally to all content in the view or to selected subsets, allowing different regions of a graph to take different layouts. The view also supports manual positioning, allowing selected entities to override the automatic layouts. Panning and zooming operations are also provided. The view may be subjected to ontological filtering, allowing the analyst to highlight, zoom and center entities of particular types. These entities may then be selected for manual positioning or for applying sublayouts, giving the analyst a number of options for organizing information in the view.

Additionally, graph theoretic functionality is available, allowing content to be added to the view based on paths, including shortest path, or vicinity queries. Either may be restricted by length as well as by included and excluded entity types. Data may be exported as a KMZ document to be viewed in Google Earth, or to the system's own Geospatial View, described below.

Geospatial View

During the course of an investigation, the analyst usually needs to document or reference locations such as countries, cities and addresses. This requires the semantic network to include such geographic location data. It may be obtained via NLP or fused to content acquired from other sources.

For example, the NER extraction of the JPG image in Figure 6 provides latitude and longitude coordinates indicating the location the photograph was taken. That geotagging metadata can be extracted from the JPG file into the network as entities and then related back to the source attachment using system-defined relationships such as "has latitude" and "has longitude," or "has geolocation" if the coordinates are combined into a single entity. Coordinates are expressed in either decimal degrees (DD) or degrees-minutes-seconds (DMS) formats (7) by default, although other formats can be supported. Once represented this way, the attachment is semantically geolocated, and the system can leverage this information in Geospatial View.

Geospatial View is a GIS-based visualization displaying geolocated and non-geolocated semantic content together. Based on the open source NASA World Wind component (8), it employs satellite imagery, aerial photography and topographic maps provided by WMS servers. The view can display information as a virtual globe, a flat-earth projection and arbitrary oblique angles. Like the system's other views, it is natively integrated into the web interface. This visualization is used to analyze geospatial relationships existing within the network. It allows analysts to examine the network in geographic terms, mapping the abstract domain of the semantic network to physical geography. This reveals and highlights locational relationships existing within the network that would be difficult or impossible to infer directly from numerical representation of geocoordinates.



Figure 10. Visualization of extracted and geolocated semantic content in Geospatial View.

Like Network View, Geospatial View presents a graph of interrelated entities. Geolocated entities and attachments are fixed to their location on the map indicated by a pointer. Non-geolocated entities related to them are positioned along the surface of the planet according to a force-directed layout algorithm. This positioning is dynamic and reactive to the panning and zooming actions of the analyst, attempting always to keep entities and relationships displayed in context to the geolocated entities at any scale, position and viewing angle. Entities may be manually positioned as well, singly or in groups.

Temporal View

Another kind of information often crucial to the analytical process is time. In order to discern chronological patterns and potential causal relationships, analysts need to be able to sequence key events and the intervals between them. This is particularly challenging when dealing with a large and complex body of information containing many kinds of events and timeframes.

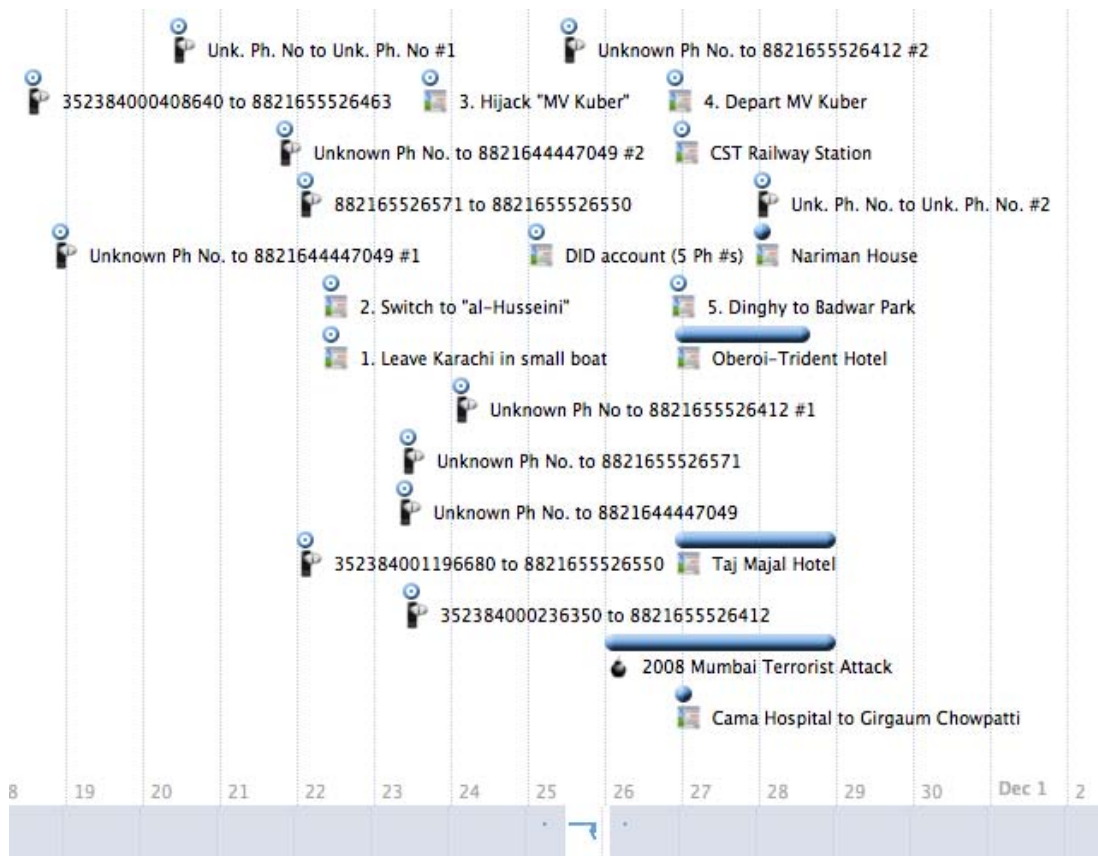


Figure 11. Temporal View.

To address this, the system includes a Temporal View to present network content as a semantically augmented timeline. In this visualization, entities and attachments are related to timestamps: entities of type Time that follow the ISO 8601 standard format (9). Depending on their relationships with timestamps, entities may be represented either as discrete milestones, having a single relationship, “has time,” with one timestamp; or as durations across a span of time, represented using two relationships (“has start time” and “has end time”) with starting and ending timestamps. The timestamps applied to entities may have been extracted or obtained via import or ingest from other sources. It is also possible to apply a timestamp to an entity manually by dragging it onto the view at the appropriate place in the timeline. Timestamps may be edited by dragging the entity to a new location on the timeline.

Temporal View indicates the entity’s type with the type’s associated icon. By hovering over the entity, the analyst accesses additional detail about the entity, options for displaying it in other views and functions for editing it or removing it from the view. The displayed timeframe may be moved forward or backward in time by dragging the timeline, and the scale of the timeline may be adjusted from seconds to millennia. If the analyst moves to a region of the timeline with no data, there are options to return by centering the view on the content and setting the scale to match the overall timeframe. As in Network View, ontological filtering is available, allowing the analyst to highlight events of particular types.

NLP Service Architecture

The NLP functionality of the system is made available to the web application via a RESTful service (10). The service layer abstracts the NLP service provider specifics, allowing the web application to access Language Now and Google Translate via a high-level common service API. Additional NLP providers may also be integrated without breaking the API contract. Depending on system configuration, MT requests are delegated to one of the two remote NLP providers. The API also gives client access to the locally integrated NER functionality.

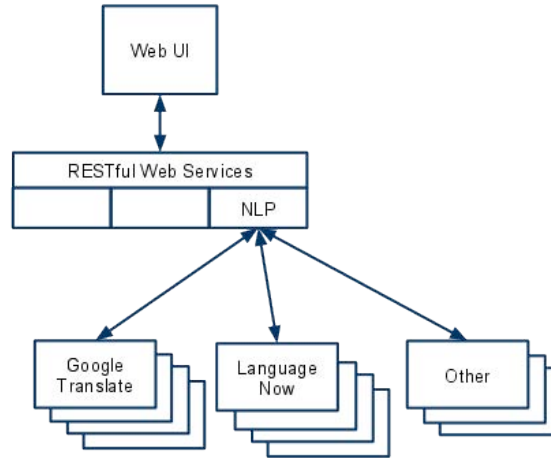


Figure 12. REST NLP service integration.

Model Integration

In addition to providing a service-independent API, the system leverages direct NLP interaction with the semantic network model, unifying NLP functionality with semantic representation. For example, the service consumer may request translation for an attachment in a network rather than having to obtain the attachment's file data and provide it directly to the service. This reduces complexity and REST overhead.

Multiple Service Configurations

The system administrator may define one or more NLP service configurations in the administration web application. These configurations may be modified at runtime without causing disruption to the analysts using the system. For example, if the administrator defined two configurations, one for Google Translate and another for Language Now, and the Google Translate configuration was active, the administrator could choose to activate Language Now instead at any time. As only one service can be active at a time, this implicitly deactivates Google Translate. The service switchover is transparent to the analyst. The only impact noticeable to the analyst is the resulting extraction itself, which can vary by NLP provider.

Multiple configurations may be defined for the same kind of provider, allowing the system to access multiple instances of the provider. This gives a manual failover capability to the administrator. If the active service is unavailable, the untranslated

text will be subjected to NER and returned to the user, making it available to the analyst for manual entity extraction.

Document Preprocessing

As indicated above, the NLP service supports a number of different file formats (25 total). Support for the broadest possible variety of file formats widens the spectrum of extraction-based data fusion and mitigates user frustration. The service also provides an API listing these supported MIME types so that the web application only offers NLP functions for the appropriate attachments in the interface. Files in binary formats are preprocessed to obtain the plain text they contain. In the case of image files, the text comes from the image metadata. For ZIP archives, the text is taken from the files within the archive. However obtained, the text is then submitted to the NLP provider for NER, MT and/or transliteration.

Rich Semantic Indexing and Search

In larger installations of Semantica Enterprise, the size of individual semantic networks can range upwards of billions of nodes or more. The total number of networks may be comparably large as well. In the semantic model, nodes may be attachments, each potentially containing hundreds or thousands of additional extractable entities, depending on the size and nature of the document content. At that scale, direct discovery of information via navigation or ontological filtering is impractical and graph analysis algorithms may not be able to produce useful results in real time. Regardless, the analyst must be able to access and obtain arbitrary semantic information quickly and as needed from large networks and diverse document corpora.

To address this need, the system provides an integrated search engine. Searches and replicated indexes are distributed across the Semantica Enterprise cloud cluster. All semantically modeled information—networks, entities and attachments, including text content contained within—is indexed at create time, making it immediately discoverable by the analyst via search queries. The system indexes uses several tokenizers, including variable length n-grams, to provide intuitive results quickly. Semantic content returned by queries may be viewed and explored in the various visualizations, imported into the active network from other networks or developed with newly created relationships with other entities.

This search capability is accessible from two places in the interface. It is available on the dashboard screen that appears immediately after logging in. In this context, outside of any particular network or ontology, the scope of the search is across all entities, attachments and networks stored in the cloud. Search is also accessible from within the analyst's active network, scoping the returned results to the network in question by default. These results may be broadened to include other networks sharing the same ontology via advanced search options. Results may be filtered by one or more entity types.

In both contexts, the system provides an “instant” search capability, in which search results are displayed and refined dynamically while typing without requiring any user action to submit the search query. The immediate feedback spares the analyst

the need to refine the search unnecessarily if the desired results are already present and allows for rapid query adjustments when needed. This feature is similar to Google Instant, which Google claims will save on average two to five seconds per search (11). This capability significantly improves the overall speed and effectiveness of searches, expediting analysis and enhancing the user experience.

Opportunities for Future Research

Over the course of this project, our investigation revealed various issues, functional gaps or weaknesses and other opportunities for improving the integrated system. Several of these areas of possible improvement, along with our recommended corrective measures, are:

- While the system can scale to networks containing very large numbers of entities and attachments, there is no way to leverage NLP functionality effectively at that scale. In addition to existing user-initiated, on-demand NLP, we recommend augmenting the system with a general-purpose bulk document ingest capability capable of importing millions of documents as attachments. An asynchronous NLP agent, implemented as a parallelizable distributed process, can extract all ingested documents and persist the extraction information prior to user request. The precomputed extractions may offer near-instantaneous response to analysts and could provide the basis for the ability to search within documents by entity and/or entity type. Adding support for Unstructured Information Management Architecture, or UIMA (12), may be of value to this investigation.
- The system's existing tools are inadequate for ingesting and managing very large document corpora. To address this, we recommend leveraging semantic tagging to organize documents. Tags serve as a mechanism for grouping collections of content and may be used to group attachments, somewhat as file directories group files on a filesystem. Unlike directories, tags are nonexclusive; an attachment may have multiple tags, allowing for less rigid, nonhierarchical document organization. This, combined with the ability to search for attachments and entities by tag and filtering by tag, would provide for much more powerful document management. The system's semantic network model currently supports tags, but a user interface optimized for managing very large numbers of documents is needed to fully leverage the system's scale.
- Currently, NER is performed after translation on foreign language documents. We may be able to improve the quality of the extraction by leveraging the translation service's pre-translation NER by default.
- The system does not fully leverage the NLP functionality provided by GATE, Language Now or Google Translate. A number of additional capabilities may be integrated, such as gazetteer lists, extractor training and co-reference resolution, thereby potentially producing richer and more accurate extractions.
- The process of manually correcting an extraction is not yet as flexible or powerful as it can be. We recommend adding the ability to split entities and to

merge extraction types by drag-and-drop. Additional investigation into simplifying and enhancing the overall user experience is likely to reveal other opportunities for improvement.

- The existing semantic network visualizations are decoupled from the extraction results until content is imported, making it difficult to know what in the network may already exist in—or relate to—information in the extraction. A new visualization can be created allowing the extracted text itself to be a view on the network. Extracted entities corresponding to entities already in the network would appear as entities do in other views, in context to other information in the same sentences. The view could make it possible to discover new sentential relationships between existing entities prior to import.

Conclusion

For this project we investigated the feasibility and effectiveness of modeling and visualizing natural language text in semantic network form. We did this primarily by exploring how NLP functionality could best be integrated into the base Semantica Enterprise system. From this investigation, we identified and engineered a number of ways to integrate semantic visualization with NLP to strengthen information analysis significantly.

We began with a proposed basic operational workflow for the integrated system, whereby analysts use NER, MT and transliteration to generate semantic content on demand. The system adopts a user-supervised paradigm to realize maximum value from imported extractions. NLP functionality is implemented within the system in a provider-neutral web service framework, configurable by the administrator, sparing the analyst the need to manage backend infrastructure.

All documents and the entities extracted from them are represented as nodes in the semantic network. The extracted information is thereby tractable to a collection of entity-centric and network-centric visualizations that portray the information in different ways: singly or collectively, as a graph or report, as well as temporally and geospatially. To provide the best user experience, all are packaged into a single unified web-based user interface.

The underlying semantic representation serves as an effective means for data fusion from different documents and formats, structured or unstructured, giving the analyst new means for unifying and analyzing complex networks of information from disparate sources. Rich semantic indexing of this content supports a distributed search engine, crucial for working with large networks and document collections.

When combined together, this diverse collection of capabilities comprises an effective general-purpose platform that empowers analysts conducting investigations. It also serves as a promising starting point for future research.

Acknowledgements

This work was supported by the CIA Office of the Chief Scientist (OCS) under award number 2010*1082127*000. The research team for this project included Tad Naworal, Mark Laffoon and David Colon of Semantic Research Inc.

Works Cited

- (1) Quillian, M. R. Semantic Memory. Ph.D. thesis. Pittsburgh, Pennsylvania: Carnegie Institute of Technology, February 1968.
- (2) GATE Project Team. GATE: General Architecture for Text Engineering. 2011. 4 March 2011 <<http://gate.ac.uk>>.
- (3) Bontcheva, K., V. Tablan, D. Maynard, and H. Cunningham. "Evolving GATE to Meet New Challenges in Language Engineering." Natural Language Engineering, 10(3/4): 349–373. Cambridge, Massachusetts: Cambridge University Press, 2004.
- (4) Maynard, D., V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. "Named Entity Recognition from Diverse Text Types." Recent Advances in Natural Language Processing 2001 Conference, 257–274. Tsigov Chark, Bulgaria: Bulgarian Academy of Sciences, 2001.
- (5) Language Now. 2011. 4 March 2011 <<http://www.language-now.com>>.
- (6) Google, Inc. Inside Google Translate. 2011. 4 March 2011 <http://translate.google.com/about/intl/en_ALL>.
- (7) Humerfelt, Sigurd. "Different degree formats: Resolutions and conversions." 2010. 4 March 2011 <http://home.online.no/~sigurdhu/Deg_formats.htm>.
- (8) NASA. World Wind. 2011. 4 March 2011 <<http://worldwind.arc.nasa.gov/index.html>>.

- (9) International Organization for Standardization. "Numeric representation of dates and time." 2011. 4 March 2011
<http://www.iso.org/iso/date_and_time_format>.
- (10) Fielding, Roy. "Architectural Styles and the Design of Network-based Software Architectures." Ph.D. thesis. Irvine, California: University of California, Irvine, 2000. Chapter 5.
- (11) Google Inc. "Search: now faster than the speed of type." 2011. 4 March 2011
<<http://googleblog.blogspot.com/2010/09/search-now-faster-than-speed-of-type.html>>.
- (12) The Apache Software Foundation. Apache UIMA. 2011. 1 April 2011
<<http://uima.apache.org>>.