CIRCA Technology

Applying Meaning to Information Management

An Applied Semantics Technical White Paper

TABLE OF CONTENTS

A. Introduction

Today's consumers, as well as knowledge workers in business and government, are inundated with massive amounts of information. While the availability of such a wealth of information provides an unprecedented opportunity for access to knowledge and the exchange of ideas, it also introduces the problem of how to organize the information in such a way that it is quickly and easily accessible.

Progress is being made to help users find information that is relevant to their immediate goals by improving search technologies and automating document classification. However, a fundamental gap still exists between the way most computer systems approach the problem of organizing information and the way in which humans wish to access that information. This gap stems from the fact that systems tend to view information (and in particular, documents) as sequences of words or numbers with no deep interrelationships, while humans approach information in terms of the meaning conveyed by words or phrases.

Humans are searching for ideas, while automated systems are limited to searching for words. Applied Semantics is working towards bringing this "human approach" into the realm of an automated system, allowing individuals to use information more effectively. Human conceptual understanding of text is driven by the wealth of knowledge that we share about how the world functions. Knowledge of entities and how they relate to each other in any given context is critical background information for making sense of what we read or listen to.

Learning from information stored in the form of text is thus a "bootstrapping" problem. We bring our background knowledge to bear on making sense of new information; this new information then becomes integrated with and augments our background knowledge, and can be used to make sense of more information. Fundamentally, it is knowledge of the concepts that words refer to and experience with those concepts that enable us to make sense of any given piece of content, which we will refer to as an "information object". It is this fact that motivates the Applied Semantics approach to making sense of textual information.

Applied Semantics' goal is to capture the knowledge that humans bring to the problem of text comprehension and apply it to enable a computer system to achieve a similar understanding. Due to its superior storage capabilities, a computer system, once given the ability to understand, will be capable of making sense of and organizing a much larger quantity of data than a human being can handle. Humans can then make use of this organization of vast quantities of information to complete our business activities and achieve our personal goals.

Our objective is not to design a so-called "expert" system that represents the sum total of human knowledge, such as the CYC system (Lenat 1995); our goal is rather to build a dynamic system that draws on some of the fundamental structuring relationships of knowledge to facilitate organization of textual information. We do not need the system to be able to reason like a human or draw precisely the same inferences from an information object as a human would in order to help someone find that information object or relate it to other information objects.

It is this strategy that enables the 3 development of a scalable, responsive system that gets at and makes use of the meanings in text, rather than simply the words.

B. CIRCA Technology Overview

The Conceptual Information Retrieval and Communication Architecture (CIRCA) is the software platform that provides access to the automated "understanding" of information that is provided by Applied Semantics' technology. Applied Semantics is dedicated to the ongoing development of this platform, while at the same time building products that leverage the technology into specific vertical markets and user applications.

At the heart of the strategy behind CIRCA rests the idea of finding connections between things that are related to one another. This can apply to any type of information object – as long as it is possible to connect the object to a specific concept or idea. That is, any object that is "about" something may be brought into the system and related to other objects. For example, a document is "about" the topics that it discusses; an advertisement is "about" the item for sale; a photograph is "about" the object(s) it depicts.

In order to determine the associations between objects, a **semantic space** is defined, within which **conceptual distances**, or the closeness of association between things may be measured. One can imagine this semantic space as a map that shows all possible concepts and topics. On this map, the concept "refrigerator" has a specific address, with the concept of "kitchen" located nearby. One would see that "dishwasher" is fairly close, compared to "umbrella", which would be much further away. The Applied Semantics Ontology, described below, defines this semantic space.

The CIRCA technology platform encompasses two major elements:

- **Interpretation of Text:** by reading text and interpreting the "meaning" of it, the system determines where to locate an object in semantic space. (We often refer to this as the **sensing** process.)
- **Semantic Search:** any object brought into the semantic space may be related to any other object therein. (We often refer to this as the **seeking** process.)

All CIRCA functionality relates to one of these elements. Many applications of CIRCA technology stem from the functionality developed for the interpretation of text, and do not necessarily rely upon the subsequent mapping of an object to semantic space for future retrieval. Other applications may rely solely on functions used for semantic searching, as the system does not depend on text interpretation in order to place objects in semantic space; they may also be manually placed.

C. Architecture of the Ontology

The Applied Semantics Ontology is the backbone of CIRCA functionality. This massive database stores the human-generated and statistical information that enables the interpretation of text.

The network of connections within this database also defines the semantic space that is used for searching by concept (i.e. idea). At its core, the Ontology consists of meanings, or concepts, and relationships between those meanings. But in order to utilize meanings and their relationships for the processing of text, we must also provide some link to the manifestation of those concepts in text, in terms of linguistic expressions such as words or phrases.

The Ontology is characterized by three main representational levels:

- Tokens: corresponding to individual word forms;
- Terms: sequences of one or more tokens that stand as meaningful units;
- Meanings: concepts.

Each term is associated with one or more meanings. Conversely, each meaning is linked to one or more terms (which can be considered synonyms with respect to that meaning). Currently, the Applied Semantics Ontology consists of more than half a million distinct tokens, over two million unique terms, and approximately half a million distinct meanings.

To illustrate the difference between the three levels, let us consider the phrase “bears witness.” This is an expression that consists of two tokens together comprising a single term, since, as a unit. These tokens have a specific usage/meaning that is not strictly a function of the meaning of the parts.

bears	witness
(Token1)	(Token2)

TERM 1 (Figure One)

In fact, this term is associated with two distinct meanings:

1. Establish the validity of something; be shown or be found to be;"This behavior bears witness to his true nature"
2. Give testimony in a court of law

Meanings are represented in the system both directly, in terms of dictionary-style glosses as shown above, and indirectly, in terms of their relationships to terms and to other meanings. That is, a concept is defined by the sets of terms that are used to express that concept, and by its location in the semantic space established through the specification of relationships among concepts.

Relationships

The types of relationships between concepts that we have chosen to represent correspond to those relationships that are fundamental to structuring human knowledge, and enabling reasoning over that knowledge. These include

- Synonymy/antonymy (e.g. “good” is an antonym of “bad”)
- Similarity (“gluttonous” is similar to “greedy”)
- Hypernymy (is a kind of / has kind) (“horse” has kind “Arabian”)
- Membership (“commissioner” is a member of “commission”)
- Metonymy (whole/part relations) (“motor vehicle” has part “clutch pedal”)
- Substance (e.g. “lumber” has substance “wood”)
- Product (e.g. “Microsoft Corporation” produces “Microsoft Access”)
- Attribute (“past”, “preceding” are attributes of “timing”)
- Causation (e.g. “travel” causes “displacement” or “motion”)
- Entailment (e.g. “buying” entails “paying”)
- Lateral bonds (concepts closely related to one another, but not in one of the other relationships, e.g. “dog” and “dog collar”)

Each relationship is associated with a strength indicating how close the relationship is.

For instance, “dog” is a kind of “pet” as well as a kind of “species.” However, the relationship between “dog” and “pet” is stronger (closer) than between “dog” and “species” and this is reflected in a larger relationship strength value. Linguistic information such as syntactic category (part of speech) and inflectional morphology (for instance, word endings indicating plurality or past tense) is associated with terms and tokens. In addition, certain meta-level classifications of tokens, that indicate how a token is used rather than specifying relationships for its meaning, are specified.

One example is identifying the language that the token is in—this identification is necessary because the ontology is organized by meanings, which are independent or outside of language. Other examples include identification of first names, trademarks, locations, abbreviations, particles, and function words.

The Applied Semantics Ontology aims to be a dynamic representation of words, their usage, and their relationships. To achieve this goal, various statistics have been incorporated into the representation of tokens, terms, and meanings, which are derived from observation of how particular words are used over a range of contexts, and with what meaning.

The probability of a specific term being used with a specific meaning, relative frequencies of different tokens and terms, the frequency of a particular multi-token sequence being used as a cohesive term, and other such statistics are gathered and used during subsequent processing. A bootstrapping methodology is followed to acquire this data, in which initial term analysis and meaning disambiguations are done on the basis of human-estimated probabilities and conceptual relationships provided in the Ontology. Statistics are gathered over this initial processing and fed back into the ontological database to be used for subsequent runs.

In addition, mechanisms for automatically generating new relationships from those represented in the basic Ontology have been implemented. These mechanisms roughly correspond to logical reasoning algorithms that infer new relationships on the basis of existing ones. For instance, given the relationships “Dalmatians are dogs” and “dogs are animals”, we can infer that “Dalmatians are animals.” Thus, relationships that are more distant are inferred from relationships that are more immediate. Using the strengths and types of the relationships on the path through the Ontology from one meaning to another, we assign a value to the strength of the newly inferred relationship.

Background

The architecture as presented here is related to a well-known semantic network called WordNet (Fellbaum, 1998), which was designed to reflect psycholinguistic and computational theories of human lexical memory. Many of the relationship types in the Applied Semantics Ontology are the same as those represented in WordNet, for the reason that those relationship types are foundational to the structure of the human lexicon.

However, the WordNet network does not include any probabilistic data, which is critical for utilizing the knowledge embodied by the network in any realistic text processing application. In addition, WordNet does not include the lateral relationships that help to organize concepts into coherent groups. This heterarchical data is central to establishing contexts that can be used to recognize particular meanings of words, since words that “go together” often do not stand in a hierarchical relationship.

D. Learning for Ontology Augmentation

In the previous section, several mechanisms for enabling the growth in semantic coverage and accuracy of the Ontology were introduced. These self-learning algorithms observe how texts pass through the system and are processed in order to generate data that feed back into the system to provide ongoing improvements in the accuracy of the data that underlie the processing. Observations of co-occurrences at the meaning level, for instance, allow the system to identify new relationships that may be important for discriminating between two meanings.

Consider the ambiguous term “Java.” This might refer to the Java programming language, an island in Indonesia, or coffee. The Ontology will represent each of these meanings as related to certain other concepts and terms; so “JavaScript” and “coding” are concepts relevant to the programming language sense of “Java”, that stand in a hierarchical relationship with that sense. In contrast, a concept such as “typing” would probably not initially be associated with that sense, as it is not in a clear ontological relationship with the meaning.

However, it is likely to be strongly indicative of that sense of the word versus the others. The association between “typing” and a particular sense of “Java” can be recognized through observation of contexts in which the concept of “typing” appears nearby the programming language sense of “Java” (e.g. “Last night I was typing in my Java code”), but not the other senses of the word. Over time, the system should recognize that the concept of typing could be used to help favor the programming language sense of “Java” over the other senses, due to their co-occurrence in particular contexts.

The relationship between the two meanings will be recorded in the ontological database, and used in subsequent analyses to bias the interpretation of “Java.” In this way, the basic ontological relationships in the Applied Semantics Ontology have been augmented with additional disambiguating relationships derived from document contexts. This effectively pulls in associations, whose strength is measured through a mutual information statistic, that come from the context of use rather than the a priori ontological analysis – **the system learns by observing meanings used in context.**

In the future, Applied Semantics will further improve the meaning analysis due to the broader knowledge base with which the system approaches the problem. The mutual information statistic is a standard metric in information retrieval literature, but in the CIRCA technology platform, it is being utilized on the level of **concepts** rather than **words**, and as such it provides a powerful basis for discriminating meanings and contexts.

E. Ontology Customization

The Applied Semantics Ontology is designed to be exceptionally broad, covering a wide range of subject areas, representing a core of “general knowledge” that is shared by humanity as a whole. While this base of knowledge is sufficient for many applications, in other situations it acts as the foundation for an additional layer of “custom” knowledge. For example, consider the internal company documents for a candy manufacturer.

Though the core of general, “common sense” knowledge applies to the information contained within these documents, there are also concepts represented that are not recognized in the base Ontology (the names of products that are manufactured by the company, for example). In addition, terms that otherwise imply common concepts may be overridden within this context (e.g., the term “Whirlwind” does not refer to a weather phenomenon, but instead, refers to “Project Whirlwind, the marketing campaign in 2001”).

The ability to “overlay” additional layers of Ontology data on top of the base Applied Semantics Ontology is built into CIRCA. This enables customization of the system to vertical contexts where greater resolution of specialized meaning is required. An intuitive user-interface provides straightforward control over this customized system behavior to the user.

In addition to creating new concepts and terms within the custom Ontology layer, it is also possible to create new types of relationships between concepts. This makes it possible for complex scenarios such as the following to take place:

- A drug manufacturer wishes to scan incoming news articles, looking for any story that refers to drugs they manufacture that cause certain side effects.
    - The manufacturer’s existing knowledge is encoded in a custom Ontology (or imported directly if already stored in ontology format):
    - Relation types “causes side effect” and “is manufactured by” are added.
    - Specific names of drugs not present in the Applied Semantics Ontology are added as additional concepts.
    - Concepts representing the drugs are linked to concepts representing side effects, by means of the new relation type. Likewise, manufacturing companies are linked to the drug concepts.
- The expanded knowledge is incorporated directly into Ontology operations. For example, the ability to intersect Ontology branches is applied in order to create a subset of drugs that share the properties “manufactured by company X” and “causes side effect stomach ache”. This set is plugged into CIRCA Detection processing in order to identify documents referring to these drugs.

This is one example of the kind of flexibility CIRCA offers when it comes to integrating into specialized domains. Accompanying the base system are professional editing tools that are in the vanguard of modern ontology-related developments.

F. Usage of the Ontology

The Applied Semantics Ontology is the database upon hich most CIRCA processing depends. Tokens, terms, and concepts stored within this database allow for text to be interpreted as a sequence of “meanings”. The relationships represented in the Ontology, and generated via inferencing, drive the processing of text at a semantic level. These relationships power word sense disambiguation, so that proper meanings can be identified, and provide a definition of semantic space, into which information objects may be placed for conceptual search and retrieval.

Pre-Processing

Word sense disambiguation technology draws on several natural language processing components as well as the data in the Ontology. Specifically, text is analyzed through the following processing stages in preparation for meaning analysis:

Tokenizer
Syntactic Category (Part of Speech) Tagger
Named Entity Recognition and Regular Pattern Identification
Term Segmenter

Figure Two

One typical natural language processing component, a separate morphological analysis stage (either stemming or more linguistic inflectional analysis), is not incorporated into this processing, as the terms in the ontological database include morphological variants, and in some case, spelling variants, of the set of different forms of terms. These were generated through automatic application of morphological rules, and reviewed by a team of lexicographers. This approach allows for an additional level of quality assurance that is not possible when “on-the-fly” stemming is performed.

Tokenizer

The Tokenizer is responsible for splitting raw data into individual tokens, and for recognizing and marking sentences. This includes handling specific formatting information represented in the input text (for instance, as might appear in HTML tags), as well as identifying specific types of tokens, such as numbers, punctuation, and words.

It maintains specific information about the original text, such as a token’s byte offset, while stripping some data out (e.g. unnecessary tags), breaking apart some white-space delimited tokens (e.g. pulling a period at the end of a sentence out into a separate token from the word it is adjacent to), or adding some structure to the input text (e.g. sentence annotations).If “spell checking” is enabled, tokens that are not identified may be matched to correctly spelled candidates, based on the statistical likelihood of the type of error.

When the input contains multiple tokens that are not clearly separated, additional token segmentation processing occurs. World Wide Web domain names are a common example of this type of input. This token segmentation process uses statistical information in order to correctly interpret the domain name “thesearch.com” as a run-together version of: “the search”, as opposed to the less likely interpretation: “these arch”.

Part of Speech Tagger

The objective of the Part of Speech Tagger is to analyze a series of tokens making up a sentence and to assign a syntactic category tag to each token. The Tagger operates on contextual rules that define possible category sequences. The tokens in the series are initialized with the most probable tags for the token as derived from the token data in the Ontology.

The tag given to each token can be changed based on the categories around that token. The part of speech data is used during the disambiguation to bias particular meanings of words. For instance, the word “branches” can be either a noun or a verb, and has different meanings in each case (“branches of a tree” vs. “The conversation branches out to …”). Knowing its part of speech in a specific context narrows down the meanings that are possible.

Named Entity Recognition and Regular Pattern Identification

The next stage of processing, *Named Entity Recognition and Regular Pattern Identification*, is responsible for identifying a series of tokens that should potentially be treated as a unit, and that can be recognized as corresponding to a specific semantic type. This module recognizes email addresses, URLs, phone numbers, and dates as well as embodying heuristics for identifying “named entities” such as personal names, locations, and company names.

Each recognized unit is marked as a term, and associated with a certain probability that the series should be treated as a unit. In the case of terms that already exist in the Ontology, this probability comes from the system’s previous observations of that term. Reference resolution also comes into play at this stage, making it possible, for example, to correctly interpret

references to “Mr. Bush” in a document, when the more ambiguous single token “Bush” is used subsequently.

Term Segmenter

The *Term Segmenter* goes through the tokens and maps single tokens or sequences of tokens to the terms represented in the Ontology. Competing terms – terms that overlap on one or more tokens – are each given a probability with respect to their competitors.

For instance, for a token sequence “kicked the bucket”, there is some probability that the phrase should be treated as a unit (a multi-token term meaning “to die”; “Grandpappy kicked the bucket last year”), and some probability that the phrase should be treated as a series of three individual terms (as in, “The toddler kicked the bucket and all the water poured out”).

As in other cases, these relative probabilities are determined by the Term Segmenter based on previous observations of those terms. Once each potential term has been identified and labeled, we have access to the potential meanings associated with each term, and the individual probabilities of those meanings relative to the term as represented in the Ontology.

When these pre-processing steps have been completed, we are left with input text that can be viewed as a series of probabilistic sets of meaning sets, where each set of meaning sets corresponds to the individual meanings of a particular term. The job of the word sense disambiguation algorithm is then to look at the context established by the juxtaposition of particular terms and meanings in the input text in order to modify the initial **context-free** probabilities of the meanings into **context-dependent** probabilities. The result of the application of the algorithm is that the intended meanings of ambiguous words in context should be assigned the highest probability.

Word Sense Disambiguation

The idea underlying the word sense disambiguation algorithm is to utilize known semantic relationships between concepts, as represented in the Ontology, to increase the probability of a particular sense of a word in context – the more words that exist in the context that are related to a particular sense of a word, the more likely that particular sense should be.

This follows from the notion of coherence in text, in that a speaker / writer will tend to use related concepts in a single context, as an idea is elaborated or relationships between entities identified. The methodology used might be described as activation spreading – each meaning in the document sends a “pulse” to the meanings close by in the document that they are related to or associated with. This pulse is used to increase the probability of those meanings.

The size of the pulse is a function of the strength of the relationship between the source concept and the target concept, the “focus” of the source concept – that is, how indicative of related concepts a concept can be considered to be (see below) – a measure of term confidence that reflects how confident the system is in the probabilities associated with the meanings of a given term, and potentially the probabilities of the source and target concepts.

The notion of focus is roughly analogous to the specificity of a concept, in that more specific concepts tend to be strongly related to a small set of things, and is somewhat inversely proportional to frequency, since more frequent concepts are less useful for discriminating particular contexts.

However, focus is not directly based on either of those notions. For instance, “Microsoft” refers to a highly specific concept that nevertheless has quite low focus, because its

presence in any given document is not highly indicative of that document being about the company or even the domain of computer technology.

On the other hand, a very rare term like “thou” also would have quite low focus – although it is rare, it does not strongly influence the interpretation of the words it appears with. We consider each of the competing terms, and each of their competing meanings, in parallel – each meaning is allowed to influence the surrounding meanings, proportional to their overall probability.

This allows meanings that may have a low a priori probability to nevertheless boost particular senses of words around it, so that the context can push a low probability meaning to the top. Several pulsing cycles, in which all the meanings in the document are allowed to spread “activation” to their related meanings, are applied in order to reach a stable state of disambiguation.

At each cycle, meanings are boosted, so that the most likely meanings are reinforced several times and end up with the highest probability. To illustrate the effect of this algorithm, consider again the example of the term “Java.”

Each of the three main senses of this term is initialized with a certain a priori probability that reflects the context-neutral probability of the term. Let’s assume that the “programming language” sense of the term is the most likely. When we look at the term in a context in which words such as “milk” and “break” appear, as shown in Figure Three, we find that only the “coffee” sense is reinforced. This is because there are no relationships between the meanings of the terms around “Java” in either the “programming language” or “island” senses.

Through the reinforcement of this meaning, and the lack of reinforcement of the other meanings, we will find that the overall probability of the “coffee” sense will become greater than the other senses. This can then be viewed as the most likely disambiguation of the term “Java” in that context.

break	java	milk
fracture	programming language	beverage
respite	coffee	take milk from mammal
good luck	island	exploit

Window of consideration
Figure Three

Sensing

After the application of the ***word sense disambiguation*** algorithm, the context-specific probability of each meaning for each term in the input text will be established. This provides a local, term-level view of the meanings conveyed by the text. However, we would also like to establish a global view of the meaning of the text – a representation of the most important concepts expressed in the text.

This has been termed **sensing**. To achieve this, the system builds on the results of the word sense disambiguation processing, again using semantic relationships as recorded in the Ontology to drive the identification of relevant concepts.

In this case, the goal is to identify the most prominent concepts in the text. This can be viewed as an analogous problem to the problem of identifying the most prominent meaning of a term, moved up from the term level to the document level. As such, the algorithm makes use of the same notion of reinforcement of meanings that the *word sense disambiguation* algorithm applies.

Related concepts that co-occur in the input text reinforce one another, becoming evidence for the importance of a concept. Implicitly, the algorithm incorporates the notion that more frequent concepts are more important, because concepts that occur more often will be reinforced more often.

But unlike many approaches to automated metatagging, this is based solely on data at the semantic level, rather than at the term level. In approaching the meaning of longer input texts, certain properties of documents are accommodated. In particular, a document may contain sections that are only tangentially related to the main topics of the document.

Consider, for example, an HTML document constructed from frames. The “sidebar” frames normally contribute little to the intended interpretation of the document in the “main” frame. Rather than handling this as a special case, it makes sense to treat this as a generic problem that can occur in documents.

This is because it often occurs that an author of a document includes something as an aside, without intending it to contribute to the main point, or because of conventions in certain domains, such as the “Acknowledgements” section of academic papers or the “Author bio” section of some magazine articles. uch sections can interfere with *sensing*, reinforcing concepts that contribute little to the overall meaning.

Furthermore, a document may contain several main points that are addressed in different sections of the document. The algorithm should identify both concepts that are important overall (i.e. recurrent themes of the document), and concepts that are discussed in depth in one portion of the document, but not reinforced in other sections.

Sensing in this case is therefore based on a view of a document as a series of regions. Each region is identified on the basis of certain heuristics, including formatting information. In general, these regions will be larger than the window considered during the sense disambiguation of any given term in the document.

Concepts within a region reinforce one another by “pulsing” across ontological relationships between them (proportional to the probability of the concept, derived from the *word sense disambiguation*, and the strength of the relationship). The most strongly reinforced concepts are selected as the most representative concepts of the region.

The representative concepts *across* regions in the document are then calculated by considering only the most representative concepts of each region, and allowing them to reinforce one another. At the end of this cycle, a ranked list of meanings important to the document will be produced. Finally, the relevance of each region to the overall meaning of the document is evaluated, and regions judged to have little relevance (because they do not contain many instances of concepts judged to be most important) are thrown out, and the representativeness of concepts across *only the remaining regions* is re-calculated.

The effect of this is that we judge the main concepts expressed in the document on the basis of only those regions of the document that seem to carry the most semantic weight with respect to those main concepts. When the text input is short, in the case of a user query to a search engine, for example, a modified form of this sense processing occurs. The “regions” of text used for longer inputs do not exist and therefore cannot be relied on to support an interpretation.

Though there is less evidence to rely on, there is a benefit to having a short input: it is possible to explore all potential meanings of an input separately. For example, when the word “Turkey” appears in a long document, one interpretation of the word will likely dominate as a result of disambiguation and sensing (say, either the bird meaning, or the country meaning). However, if the entire input text is the word “Turkey”, the ambiguity of the input can be preserved, and both interpretations passed on to subsequent processing.

The huge number of possible permutations of meaning for longer documents makes this infeasible when they are used as the input. At the end of the process, we have a ranked list of meanings most representative of the document. This is the “gist” of the document, and represents the document’s location in the “semantic space” that is used for semantic search and retrieval.

G. Applications of CIRCA Technology

How is the power of the CIRCA platform harnessed in Applied Semantics products? All Applied Semantics products are based on CIRCA’s ability to interpret the meaning of text (i.e. “sensing”) and/or our ability to determine the relatedness of objects that have been interpreted (i.e. “seeking”). Applied Semantics products and services utilize CIRCA technology for processing text, from our domain name and online advertising web services to our enterprise-scale categorization solutions.

CIRCA software has been designed to be extremely open and easy to integrate into a wide variety of environments. The software runs on Windows, Linux, or Unix systems and performs all communications via an XML format over HTTP. This interface allows for any programming language to talk to the system; in addition, a C and C++ API is available.

These are but a **few** of the applications for CIRCA technology that today enable users to transform unstructured information into structured, actionable knowledge and thereby unlock new sources of revenue and cost savings.

Document Categorization

Categorization of a document into a category (or set of categories) requires recognizing the main topics addressed in a document and finding the categories that best correspond to those topics. Automatic conceptual categorization of documents, a capability in our Concept Server and News Series products, offers many benefits including but not limited to:

- Improves overall organization of unstructured data
- Reduces the costs of manually assigning categories to content
- Dramatically improves applications and business processes involved with browsing, searching, accessing, and retrieving content
- Enables unstructured data to conform to standard taxonomies
- Facilitates information sharing – inside and outside an organization

Because our categorization model is based on semantic relationships, rather than term frequency like many alternative solutions, it does not require training data in the form of huge

numbers of pre-categorized documents. It works solely from an understanding of the categories, an understanding of the documents, and the ontological relationships they share.

To complement automatic conceptual categorization, Applied Semantics has also developed a taxonomy administration tool that allows users to:

- Edit existing taxonomies and/or create new ones
- Set up or use one or more taxonomies imultaneously
- Map categories to concepts that best represent them
- Automatically determine concepts that are most representative of a category by submitting text or sample document.

Online Advertising

Another application of CIRCA is our suite of Web services offerings that use our semantic searching capability to associate various types of input content with advertisements that are conceptually related. AdSense, our online advertising product, uses this functionality, to deliver a contextually targeted advertising solution to online publishers and web portals.

Online marketers are searching for new ways to engage consumers, especially as users increasingly tend to tune out traditional online ad formats. Advertisers need a means to increase the relevancy of their campaigns by forging innovative alliances with content publishers and targeting ads based on the content of web pages and the interests of users.

Contextual targeting ad technologies extract the key theme(s) from a web page in order to target an ad about the same topic. Unlike demographic profiling, which requires prior knowledge about users, contextual targeting works by implicitly identifying an individual user's interest based on the key theme(s) of the web page content he is reading in real-time.

Research has shown that users are ten times more likely to click on a contextually targeted ad versus a traditional run-of-site ad, proving that targeting not only provides great value to advertisers but also provides content that is of interest to users.

AdSense enables online publishers and web portals to:

- Better monetize their large base of users
- Better monetize undersold advertising inventory
- Achieve high levels of relevancy and targeting for ad content
- Ultimately garner higher click-through rates and CPMs (cost-per-impression) to increase their revenue

AdSense is truly a revolutionary, turnkey solution that bridges the world of content, technology, and advertising.

Domain Name Industry

Applied Semantics supports a number of products centered on Internet domain names. Application of CIRCA in these cases is similar to those already described – the main difference being simply the type of input and/or output involved in processing. When it comes to understanding domain names, the ability to segment tokens properly becomes extremely important (as in the "thesearch.com" example described earlier).

Some applications of CIRCA in the domain name industry include:

- DomainSense: A domain name generation service that suggests conceptually related domain name alternatives to maximize revenue for domain name registrars.
- DomainPark: A service that monetizes traffic to parked ("under construction") domain names by mapping a domain name to related advertisements.
- DomainSearch: A service that analyzes an input domain name in order to find other domain names that are conceptually related, helping users to find sites while browsing, or to purchase domain names that are for sale.

Document Processing for Information Extraction

Several important applications of CIRCA technology stem from the natural language processing that is performed on input documents. These applications may not employ a "semantic seek", in which an input is conceptually compared to other objects, but do take advantage of most of the same semantic analysis. By being able to recognize, add to, and extract information from documents, users can deploy many valuable new applications that unlock the value in their corporate content.

Subject metatagging involves the identification of terms that are descriptive of a document and that can be used by other software systems for the indexing and retrieval of that document. Once we have sensed a document, we have a collection of the most important concepts for that document. In our Ontology, each concept is directly associated with one or more terms.

Furthermore, each concept is ontologically related to other concepts that are also directly associated to terms. At its most basic level, identifying metatags is simply returning the terms in the document associated with the most important concepts. More complex output processing is possible by extending usage of the Ontology. For example, synonyms and other closely related concepts may be pulled in to create more comprehensive metatag sets.

Summarization of text, that is, identifying portions of a document that can be extracted and used as effective summaries, is another important benefit of the natural language processing performed within CIRCA. In generating a summary for a document, the aim is to present information that conveys the main points of the document, in order to allow a user to quickly review the document and decide whether it is interesting to them or relevant to their goals.

The Applied Semantics summarization tool generates a summary by extracting the most representative sentences from a document. The algorithm for selecting those sentences builds off the sensing results for the document. It looks for sentences in the document that contain many concepts that match or are closely related to the most important page senses of the document.

The algorithm is weighted to prefer sentences that have a broader coverage of the important meanings, although a sentence that strongly references only one of the important meanings can certainly be selected. Sentences that include concepts which match the senses directly are also weighted more highly than those that include concepts related (as represented in the Ontology) to the senses.

Each sentence in the document is given a "sense match" score according to these criteria. The final product is a ranked assessment of the sentences relative to the senses. It is presented to the user according to specified options on sentence length (a specific fixed length, or a certain percentage of the document), and sentence order (in order of match rank, or in order of appearance in the document).

Summarization and subject metatagging are considered to be two specific instances of the process we refer to as **Detection**. This refers to the general problem of searching for *details*

of some kind *within* a document, as opposed to working with the gist, or overall meaning of the document, as occurs with a semantic seek operation.

Detection is implemented within CIRCA as an extremely customizable feature that allows for the creation of independent targeting rules and extraction rules. Targeting rules determine what is to be identified within a document. For example, a rule requiring that the name of a “software company” must appear in a sentence might be combined with another rule requiring that the concept of “buying” appear. (Accessibility of data stored in the Ontology makes defining these rules as simple as this actual description.)

This combined targeting rule may be employed by an extraction rule that specifies the way in which text is extracted from the original document, and what kind of markup on the original text should be used to identify the discovered targets.

H. Conclusion

Through the approaches outlined in this paper, Applied Semantics is creating exciting new natural language processing technology and software products that extract the concepts residing within any type of text-based information, from short text strings to longer documents such as Web pages, much the same way humans make sense of information. We have harnessed this ability to conceptually map information into a variety of industry-specific applications that solve real-world problems.

References

1. Fellbaum, C. (1998)."WordNet: An Electronic Lexical Database". MIT Press, 423 pp.
2. Lenat, D. B. (1995). "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11, November 1995.s