

Ontology Usage and Applications

Applied Semantics Technical *White Paper*



TABLE OF CONTENTS

A.	Introduction	3
B.	Architecture of the Ontology	4
C.	Learning for Ontology Augmentation	7
D.	How the Ontology is Used	8
E.	Applications	13
F.	Conclusion	15

A. Introduction

Human beings today 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 cross-fertilization of ideas, it also introduces the problem of how to organize the information in such a way that the information is easily digestible and accessible. Great efforts are being made to facilitate the process of helping users to find information that is relevant to their goals, by improving search technologies and automating document classification.

However, a fundamental discrepancy still exists between the way most automated systems approach the problem of organizing data and the way in which humans wish to access that data. In general, systems view the data (and in particular documents) as sequences of words or numbers with no deeper interrelationships, while humans approach the data 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 ideas into the realm of an automated system, building tools that make sense of data at a level much closer to human understanding, in order to facilitate the achievement of human goals with respect to that data.

Human conceptual understanding of text is driven by the wealth of knowledge that we share about how the world functions. Knowledge of entities and relationships between them is critical background information for making sense of text. Learning from information in the form of documents 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 then 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 a human to make sense of text.

It is this fact that motivates the Applied Semantics approach to making sense of textual data. We wish 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. Once the computer system is given the ability to understand, it will be capable of making sense of and organizing a much larger quantity of data than any single human being can handle, due to its superior storage capabilities. A human can then make use of this organization to achieve personal goals and to filter the quantity of information down to a manageable level.

Our objective is not to design an 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 that knowledge to facilitate organization of textual data. We do not need to be able to reason exactly like a human or draw precisely the same inferences as a human would from a text in order to help a human find a document or relate it to other documents. It is this strategy that enables the development of a scalable, fast system that nevertheless gets at and makes use of the meanings in text, rather than simply the words.

The primary mechanism for implementing this strategy in our Conceptual Information Retrieval and Communication Architecture (C.I.R.C.A.) technology is the vast Applied Semantics Ontology.

B. Architecture of the Ontology

At its core, the Applied Semantics Ontology consists of meanings, or concepts, and relationships between those meanings. But in order to utilize meanings and their relationships while processing 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 therefore is characterized by three main representational levels:

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

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 Ontology consists of close to 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.

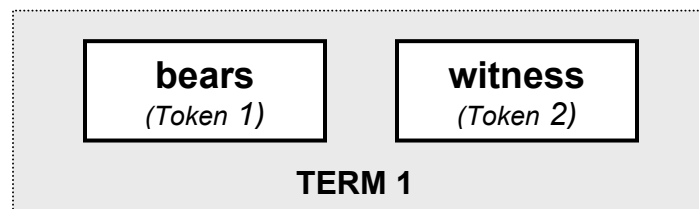


Figure One

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.

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. We represent:

- Synonymy/antonymy (“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/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 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.

To make use of enterprise-specific or otherwise pre-existing categories or domain-specific taxonomies, the system supports the linking of external terms with the meanings in the Ontology. This allows the results of any semantic analysis done by the system to be mapped into proprietary or pre-existing classifications, essentially providing the external terms with hooks into the massive knowledge base represented by the Ontology and giving those terms meaning, independent of the specific context they were developed for.

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.

C. 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 “swing classes” 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 can 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 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. On the next iteration, we are likely to further improve the meaning analysis due to the broader knowledge base with which the system approaches the problem. Note that the mutual information statistic is a standard metric in the information retrieval literature, but that 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.

D. How the Ontology is Used

The relationships represented in the Ontology, and generated via inferencing, drive the processing of texts at a semantic level. The primary use of the Ontology is for word sense disambiguation, which in turn provides the foundation for document categorization, meta-tagging, and summarization.

Pre-Processing

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

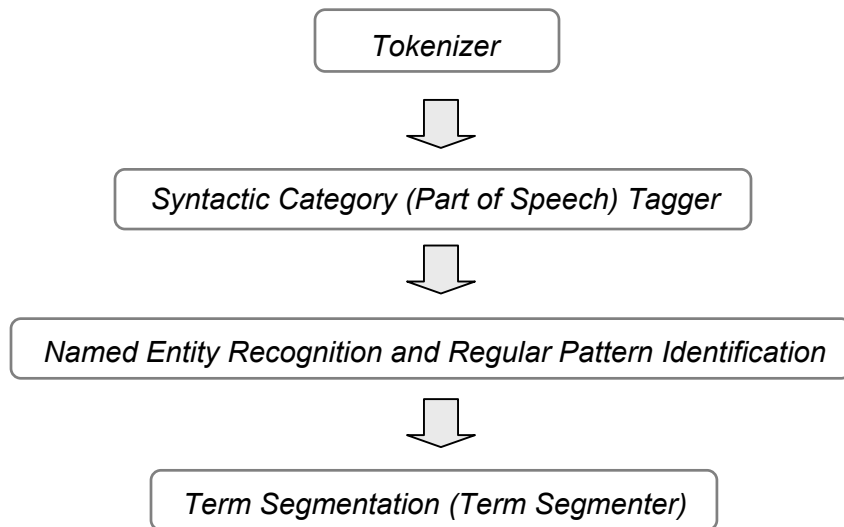


Figure Two

Note that 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 lemma terms. These were generated through automatic application of morphological rules, and reviewed by a team of lexicographers.

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 documents (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 document, 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 document (e.g. sentence annotations).

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 current *Tagger* is implemented as a finite state transducer based 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 by the finite state machine 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 what meanings are possible.

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.

The *Term Segmenter* goes through the tokens and maps single tokens or sequences of tokens to the terms represented in the ontological database. 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”). These relative probabilities are determined by the *Term Segmenter*, again, 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 ontological database.

When these pre-processing steps have been completed, we are left with a document 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 a single document, 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 boost 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 analagous to the specificity of a concept, in that more specific concepts tend to be strongly related to a small set of things, and is roughly inversely proportional to frequency, since more frequent concepts are less useful for discriminating particular contexts. However, it 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 (page 11), 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.

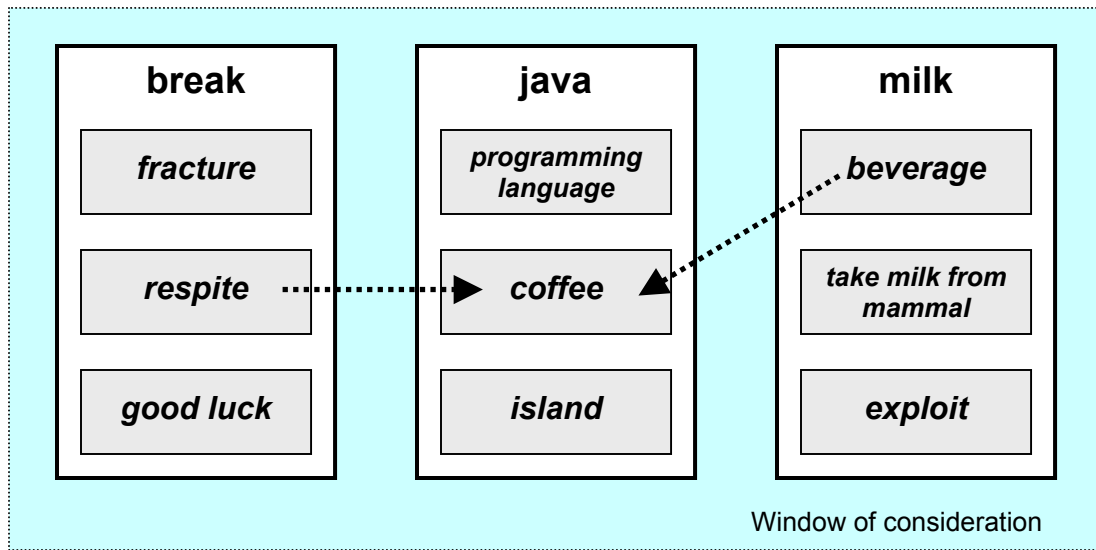


Figure Three

Sensing

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

In this case, the goal is to identify the most prominent concepts in the document. 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 document 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 meta-tagging, this is based solely on data at the semantic level, rather than at the term level.

In approaching the meaning of the document as a whole, 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, however, 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.

Such 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 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.

Then, the representative concepts *across* regions in the document are 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.

At the end of the process, we have a ranked list of meanings most representative of the document.

E. Applications

How do we harness the power of this semantic-level processing for the Applied Semantics products? Each application of the technology builds on the results of *sensing*. They will be briefly discussed here.

Meta-tagging

The goal of meta-tagging is to identify terms that are descriptive of a document and that can be used 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. Therefore, identifying meta-tags at its most basic is simply returning the terms in the document associated with the most important concepts.

However, for more discriminating sets of meta-tags, we probably wish to expand upon this basic set of terms derived directly from the document. We can expand in two different ways:

1. Inclusion of synonymous terms – that is, we may add terms to the set of meta-tags that do not explicitly appear in the document, but that express the same concept.
2. Inclusion of terms derived from related concepts – we may utilize the relationships in the Ontology to identify strongly related concepts, and add the terms corresponding to those related concepts into the set. For instance, if a document is strongly about the coffee sense of “Java”, we may wish to add “coffee” to the set of meta-tags, even if “coffee” is never explicitly mentioned in the document.

Furthermore, in the construction of the sets of meta-tags, there are a few options:

1. Focus on a narrow set of concepts from *sensing*; that is, pick only the top 1 or 2 highest ranked concepts. This could be construed as “the main topic” of the document.
2. Include a broader set of concepts from the ranked list of concepts derived from *sensing*. In this case, we allow secondary topics to appear in the meta-tags.

These two sets of options correspond to two different dimensions affecting the construction of meta-tag sets, that can be manipulated independently.

In the context of using these meta-tags for document retrieval, the first dimension will mostly impact recall – the system will retrieve more relevant documents, as it will incorporate documents mentioning a broader range of concepts. The second dimension will mostly impact precision – the system will be more likely to retrieve only relevant documents if we focus on documents that mention a more narrow range of important

concepts. Thus, having the capability of manipulating the two dimensions would potentially allow a user to customize the precision/recall balance that they are interested in.

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.

In line with this, the Applied Semantics categorization technology is based on the idea of mapping the results of document *sensing* into category definitions. Category definitions in this case are mappings from categories into weighted sets of concepts drawn from the Ontology. Each of these category definitions can be thought of as a vector in high-dimensional semantic space established by the Ontology, and the relationships recorded there.

The process of identifying the categories that are relevant to a document takes the weighted collection of concepts generated by *sensing*, and for each possible category, evaluates the closeness of that collection and the weighted set of concepts representing the category. This closeness is a proprietary measurement which intuitively represents how strongly the document meanings project onto the meaning expressed by the category – how much semantic overlap is there between them? In this way the system identifies the categories for which the document is representative.

This same process can be used in the context of search and retrieval: in this case we think of a document as occupying a certain location in semantic space; for any disambiguated query, we attempt to map it onto the documents which are closest in meaning.

Note that this view of categorization, and the corresponding view of search, is quite unique: it is a model based on semantic relations, rather than on term frequency. Furthermore, 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.

Summarization

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 (for example) 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 of 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).

F. Conclusion

Through the methodology outlined in this paper, Applied Semantics is creating natural language processing technologies to extract the concepts residing within documents, much the same way humans make sense of text, and apply this conceptual information toward making information retrieval a simpler and more successful process. The use of a vast ontology capturing the core relationships between concepts supports the characterization of text in terms of *meanings*. Documents and the words they contain are placed into semantic space by harnessing those relationships and recognizing the conceptual coherence of text. Each document processed by the system can then also serve as evidence for further, new conceptual relationships, creating a sophisticated and dynamic framework for making sense of texts.

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.