Semantic Search: Content vs. Formalism

Christian F. Hempelmann[1], Victor Raskin[1,2]

[1]hakia.com, New York, NY, United States
[2]Purdue University, West Lafayette, IN, United States
chempelmann@hakia.com, vraskin@purdue.edu

Abstract

This paper presents a theoretical approach for deep-meaning representation, ontological semantics (OntoSem), for a specific, complex NLP application: a meaning-based Internet search engine. It introduces the resources and technologies of OntoSem and their development into OntoSem2. The aim is to provide a general overview of the specific methods in which OntoSem is used in our Internet search approach and give an in-depth account of selected key issues in web search and how we address them. OntoSem2 parses natural language web content and transposes it into a representation of its meaning, structured around the events described in the text and their participants. Queries can then be matched to this meaning representation in anticipation of any of the permutations in which they can surface in written text. These permutations centrally include overspecification (e.g., not listing all synonyms, which non-semantic search engines require their users to do) and, more importantly, underspecification (as language does in principle). For the latter case, ambiguity can only be reduced by giving the search engine what humans use for disambiguation, namely knowledge of the world as represented in an ontology. One central issue around which the paper will be structured rhetorically is the distinction between semantic content and purportedly semantic formalisms. Meaning for web search requires complex description for automatic generation and can in principle not be extracted from surface text with statistical methods. In contrast to this, formalisms and suggestions for controlled vocabularies like OWL may claim to be semantic, but can, of course, not be, since meaning is content and does not lend itself to automatic extraction from natural language without rich knowledge resources.
Index Terms: ontological semantics, internet search

1. Introduction

Semantics can be done semantically or with the method *du jour* in order to avoid having to do it semantically. These methods, largely statistical and formal-logical, tend to hide the lacking motivation for their application to an issue that requires access to meaning beyond neat formalism. Jackendoff specifically addresses this problem with respect to Latent Semantic Analysis (LSA; Deerwester et al. 2000), a prominent non-semantic tool to avoid doing semantics: "We cannot afford the strategy that regrettably seems endemic in the cognitive sciences: one discovers a new tool, decides it is the only tool needed, and, in an act of academic (and funding) territoriality, loudly proclaims the superiority of this tool over all others" (2002: xiii).

Doing semantics semantically, on the other hand, means being aware of the importance of meaning determination for the processed text. This awareness is universally shared since funding for non-semantic projects evaporated in the mid-1990s. hakia's team is part of a small minority who "does it semantically" while the vast majority "does it non-semantically".

Practically, doing semantics semantically means to emulate human processing. This entails to acquire massive human-like knowledge resources, in particular a language-independent ontology (conceptual hierarchy) and a language-specific lexicon (anchored in the ontology). It means to acknowledge the compositional basis of sentence meaning and an aspiration to more than 95% accuracy, because less is not acceptable to human users. Ultimately, it means the implementation of systems based on these resources as the only valid evaluation criterion.

If you don't want to do semantics semantically, for reasons briefly speculated about below, you would usually use syntactic/statistical/tagging/annotating methods in order to not have to acquire any semantic resources. In other words, your aim is to guess meaning from non-meaning phenomena, like co-occurrence and other surface structure properties of language. Such attempts to get to meaning usually based on co-occurrence and its quantification are often hailed as non-aprioristic and empirical, whereas they should rather be exposed as unwittingly non-theoretical, which entails non-scalability and usually non-implementability: "[D]o the data of performance exhaust the domain of interest to the linguist, or is he also concerned with other facts, in particular those pertaining to the deeper systems that underlie behavior? [This] behaviorist position is not an arguable matter. It is simply an expression of lack of interest in theory and explanation. [...] Characteristically, this lack of interest in linguistic theory expresses itself in the proposals to limit the term 'theory' to 'summary of data [...]" (Chomsky 1965: 193f).

Statisticians who claim to do computational linguistics or psychology or any other task where quantification is assumed to be a heuristic method, often adorn their work with this quote by Kelvin (1889): "When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science." We think Kelvin mistook quantifiable method for science, as Nietzsche observed: "It is not the victory of science that characterizes our 19th century, but the victory of the scientific method over science" (our translation; Nietzsche 1901: 466). For semantics, Lyons dispatched this line of reasoning: "Not all that is measurable is meaning!" (1963: 13). But meaning is what linguistics is about.

Practically, if you don't want to do semantics semantically, you will emphasize the formality and formalisms of precise, quantitative methods, achieve and accept less than 85% accuracy, get excited about 0.028% improvements, because you hardly ever implement real-life systems and replace them with artificial self-serving criteria of evaluation (e.g., Senseval "competitions").

The reasons why semantics is often not done semantically include a lack of qualified descriptive linguists, a lack of semantic preparation of others beyond "formal semantics," the confusion of formality with formalisms, and a devotion to legacy methodology of the pre-semantic period, on the one hand. On the other hand, there is a lack of NLP preparation in linguistics, no notion of theory-based methodology and no understanding of the user's high acceptance threshold: 85% is no good, and neither is 85.028%.

2. Ontologies: Content vs. Form

Without any experience in linguistic description, it's easy to confuse a formalism with what it is supposed to formalize: It is one thing to develop a formal environment for accommodating certain types of information. It is an entirely different task to make that information flow into this environment. Most US funders order a working system but get an environment fully supported by quality software—and nowhere near a working system. Along these lines, the Semantic Web hype is currently running or might already have run its course. Web Ontology Language (OWL) is an elaborate knowledge representation environment that comes without methodology, concern, or understanding of how the information will automatically flow from text into OWL: What are the theories behind the grammar?

Besides the above-mentioned attempts to simulate meaning analysis with statistics and/or using the ineffective WordNet (Fellbaum 1998) resource for the same purpose, an increasing number of search engine companies have been trying to make semantic claims. Further scrutiny of these claims discovers the absence of the necessary resources or know-how for meaning representation, and the claim is reduced at best to the use of a crude taxonomy or a quasi-ontology, with an occasional reference to the Semantic Web (Berners-Lee et al. 2001). The ontological semantic resources and technology are essential for search and other advanced applications because OntoSem2 actually delivers what the Semantic Web initiative and various government-supported and commercial ontological projects promise to develop some time "down the road." The Semantic Web has not made any real progress towards its vague goal of making the semantic content of the websites available to all because they have focused exclusively on the development of a complex formalism, OWL (700 pages worth of rules) for expressing the content potentially rather than on the methodology of automatic flow of text and data information into that formalism. The various ontological products, especially, the government-supported ones, are old taxonomies under a new name, often enriched with a second or third level of "children."

In his famous comment on NLP-based search technology, Sullivan (2005), the owner of the influential site SearchEngineWatch.com, correctly criticized a number of companies making NLP claims for not delivering much by way of real quality. Unfamiliar with our technology, he proceeded to dismiss any NLP technology, saying, correctly, that to be effective, the system must know every word in the lexicon, and that, he added incorrectly, is decades down the line. Knowing every word in the lexicon is just one part of our approach, and it is here and now, along with much more to make real meaning analysis in the search a commercial reality.

The different uses of "ontology" in semantic and non-semantic NLP illustrate the distinction between form(alism) and content well.[1] The word "ontology" is currently used to denote at least three distinctly different kinds of resources that have distinctly different kinds of uses, not all within the realm of NLP and text processing. Obrst (2007) includes a nice graph of the "ontology spectrum," which relates the different kinds of databases to each other. A large percentage of ontologies are controlled vocabularies, organized as taxonomies or thesauri. These are not really "ontologies" in any sense of the word because they contain no or very little relational information between concepts. They are useful for establishing standard usage of vocabularies and other pieces of information, and organizing, sorting, and modifying databases. These ontologies can grow to enormous sizes of hundreds of thousands or even millions of pieces of data because they have no mechanism for cross-categorizing and specifying within each datum. In this way, they stand in contrast to the true ontologies, which contain conceptual information, meaning that each individual entry is no longer a single datum, but rather a compilation of data about a thing. Ontologies of this sort are able to relate entries to each other in a variety of ways and to make cross-comparisons of the properties of their entries. Because entries are more complex, they are likely to contain far fewer entries, probably on the order of singles or tens of thousands. OntoSem leverages this kind of "weak ontology" for NLP by linking it to a controlled vocabulary (a lexicon) for each language that the system is competent to deal with. Though the ontology in ontological semantics need be very small–possibly fewer than 10,000 concepts–the lexicon for each language is as expansive as one would expect a controlled vocabulary to be, with the English lexicon currently well over 100,000 terms, and not having moved extensively into domain specialization.

The final stage of evolution in ontologies is the "strong ontology" in which entries contain information that allows for some kind of inline computation, usually first order logic or mathematical equations. Because the OntoSem approach appreciates flexibility and because it is geared towards NLP, in which any given rule can usually be found to have exceptions, it has chosen to place calculations and logic into whatever software components may use the ontology and lexicon, rather than into the ontology itself.

The following bullets and related brief comments below capture the major differences between the ontological semantics ontology-cum-lexicon and a typical controlled-vocabulary type:

- full meaning representation, in ontological terms, of each lexical entry;
- nodes with interrelated properties: as shown below, each concept is a set of ontological properties;
- properties as essential part of ontology;
- non-monotonic inheritance;
- domain extension capability;
- fully automatic applications: going far beyond the noble task of terminology unification into a large set of meaning processing applications (see the last section) and on to non-natural-language ontological support for specific domains;
- focus on content, not formalism: results in easy compatibility with any reasonable formalism.

The last bullet point is of crucial importance here. There is a strong tendency, in the formalism-oriented communities, including this one, to confuse the knowledge representation formalism, which provides the means to capture some information, with the final results of NL or data information

[1] We owe this section to Triezenberg's contribution in Raskin *et al.* 2008.

having been actually transposed into that format. Most of the work stops at the formalism, and most of the discussion is about "my formalism being better than yours." There is nothing revealing about our formalism—it is just your household LISP format. In fact, we do not care about formalism, and our resources can be—and have been (usually unnecessarily)—easily transposed into any format. It is what the formalism stands for that makes this approach unique—content, not form. It stands for natural language meaning as understood by humans, and it comes with the technology for automatically transposing NL text or data into the ontology-based formalism.

3. Implementation

A second dimension, theoretically orthogonal to the degree to which linguistics informs computational linguistics, but correlating to it because of similar psychological and disciplinary-philosophical issues, is the degree to which an NLP system is really (intended to be) doing something useful. Because statistical systems hit a ceiling of performance below levels of user acceptability, they cannot be employed for a real task. Seemingly easy, they are stop-gap measures that don't scale up and the non-scalability is usually overlooked, because the systems are not implemented. So they are models of systems that would not be feasible to be built at a real scale, because the feeble materials they are built from could not bear the necessary loads. From this stems the prevalent culture in NLP of creating proofs-of-concept and toy applications in limited domains, as well as comparing these proofs-of-concept and toy applications to each other. While they are indeed comparable to each other, the performance data derived in that way have no meaning, because they don't reflect performance in relation to an implementation.

Actually using software provides criteria for its design and a metric for its success or failure. Implementedness puts the following main demands on a system (cf. Hempelmann 2007):

- Knowing on what input you can and can't fail: non-critical gaps, allowing for low-penalty errors.
- Being able to handle unattested input: robustness.
- Operating fast enough for users to accept the implementation: speed.
- Having a theory how this method can in principle be improved: scalability.

4. Implemented Content-Oriented Semantics: Ontological Semantics

Ontological semantics developed from early work in what later became known as computational semantics, in the late 1960s-early 1970s at Moscow State University, on meaning-based NLP systems for limited domains/sublanguages for science and technology (Raskin 1971), development of script-based semantics in the early 1980s (Raskin 1986), and concurrent work on the semantic interlingua (e.g., Nirenburg, Raskin, and Tucker 1987). After a joint NSF grant, shared by the Purdue NLP Lab (PNLPL) and the Center for Machine Translation at Carnegie Mellon University, the initial, "legacy" set of ontological semantics resources was created (see Nirenburg and Raskin 2004), primarily in the MikroKosmos MT project, largely at the Computing Research Laboratory (CRL) at New Mexico State University in Las Cruces, NM, in 1994-2000, drawing also on student-labor and research support from PNLPL and, increasingly since 1999, on the involvement of the Purdue Center for Education and Research in Information Assurance and Security (CERIAS) in extending the legacy resources to the domain of information security and its subdomains.

Since 2004, there has been a massive ontological and lexical acquisition effort guided by the intended application for Internet search. Proprietary software for the meaning-based search engine was created, to the total elimination and replacement of the legacy resources, resulting in OntoSem2.

Our current inventory includes the following resources:

- a language independent ontology with ca. 7,000 concepts,
- several ontology-based lexicons, including a 50,000-entry English lexicon with 80,000 senses, and a several smaller lexicons for other languages,
- onomastica, dictionaries of proper names for several languages; the current one with ca. 20,000 entries and a total of 25,000 senses,
- a text meaning representation (TMR) language, an ontology-based knowledge representation language for natural language meaning,
- a fact repository, containing the growing number of implemented TMRs,
- a preprocessor analyzing pre-semantic (ecological, morphological, and syntactic) information,
- an analyzer (ontological parser) transforming text into TMRs, and

An ontological semantic system represents input text as a complex TMR—initially, one for each clause. In other words, ontological semantics has developed the ability to represent the meaning of text automatically, thus emulating the mental processes of a human who reads a text. Very simplistically, the OntoParser reads every sentence linearly, looks up every word in it, and gets to the underlying concept(s) in the semantic structure (sem-struc) of each lexical entry.

The OntoParser output is stored in the Fact Repository. Because the first example, like most sentences in NL, does not match all the numerous properties of the event concept, other sentences of the same text, upon successful co-reference resolution, may provide additional information that will fill the slots in those additional properties.

In reality, things are much more complex, as they always get when dealing with the facts of reality and NL reflecting and underdetermining it. The identification of the main even may get hard. There may be competition for the essential slots. There may be no filler for an essential slot, such as agent or theme. Some techniques to resolve these and many other issues are dealt with in Nirenburg and Raskin (2004, Ch. 8); many more have been developed for the proprietary OntoParser, under development at hakia.com.

The success of semantic search closely correlates to the stage of OntoParser implementation and thus, is expected to improve constantly. Fewer problems occur because of unattested income, e.g., the occurrence of a word which is not in the lexicon, because the system is very robust in guessing and even automatically creating a partial lexical entry for such a word (ibid.), but they can never be excluded entirely. So while the proposed system is, in a sense, a work in progress—as most research usually is—the current stage in the development of the OntoParser can already support a useful system.

Ontology in the approach is a constructed model of reality, a theory of the world. It is a highly structured system of concepts covering the processes, objects, and properties of a domain in all of their pertinent complex relations, to the grain size determined by an application or considerations of computational complexity. Thus, an ontology may divide the root concept as shown in Figure 13; EVENTS as in Figure 14; OBJECTS as in Figure 15; PROPERTIES as in Figure 16.

Figure 1: *The top-level of the ontology.*

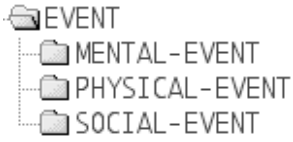

Figure 2: *Top-level of the EVENT-branch of the ontology.*

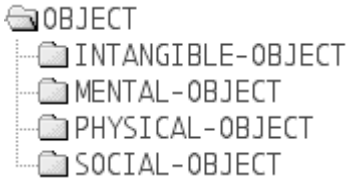

Figure 3: *Top-level of the OBJECT-branch of the ontology.*

Figure 4: *Top-level of the PROPERTY-branch of the ontology.*

Formally, then, an ontology is a tangled hierarchy of conceptual nodes, each of which can be represented as:

(1) concept-name
(property-slot property-value)+

In other words, a concept has one or (usually) more properties. Every concept but the root ALL has the property IS-A, and the value of the property is the parent of this concept, the higher node—so the concept MENTAL-PROCESS, a child of PROCESS, is, on partial view, as follows:

(2) mental-process
is-a process
(property-slot property-value)+

Equipped with this capacity of representing the meaning of input text as a TMR, the approach has been successfully implemented in a couple of dozen applications, ranging from the highly accurate MT to IR, IE, DM, and QA. It has also developed a new toolbox for homogeneous semi-automatic acquisition of ontological concepts and lexical entries, a vast improvement on the slow and defunct KBAE (2002).

5. Semantic Semantics Implemented for Internet Search: OntoSem2

The goals of the symbiotic effort to use OntoSem for Internet search, uniting bag-of-word (BOW) shortcuts and OntoSem, have been as follows:

- to differentiate between the use of full-fledged OntoSem resources for offline operations in Internet search, namely crawling, semantic parsing, and storage and retrieval of the parsing results, and
- to develop a battery of quick, cheap incremental enhancements to each current phase of the search engine development which permanently, dynamically, and asymptotically move the product to the optimal meaning representation at runtime.

The performance of software is in part determined by the deftness with which the user operates it. A user who understands the inner workings of a program can manipulate it to perform in the way the user desires, and efficiently extract results; a user who is new to the software and unfamiliar with the unforgiving algorithmic logic of computers will likely have a lot of initial trouble getting any results at all. Users have become conditioned over time to deal with BOW-based search engines, and tweak their queries by reducing natural-language questions to only nouns and verbs, inserting Boolean operators and punctuation commands, and especially making multiple searches for synonyms.

The following discussion will outline several of the measures that OntoSem allows hakia to make Internet search a venture in which the machine cooperates with the user on their ground of natural language rather than forcing them to into their realm of artificiality in the worst sense.

5.1. Finding Parallel and More General Matches

OntoSem makes it possible for the naïve user (as well as the experienced one) to achieve optimum search results with a single search and no "tweaking" of their search terms. For example, suppose that a user with a pounding headache wants to know what remedies for headaches are available and appropriate for him or her. A BOW search for this information might be "aspirin headache," or "cure headache," and neither would produce all of the desired results. Our OntoSem search engine, on the other hand, takes the natural language query "does aspirin cure headaches?" and automatically expands upon the query to produce a thorough search. "Aspirin" would trigger a search not just for the word aspirin, but rather for all words linked to its ontology concept, and words linked to that concept's parent and child concepts—not only "aspirin" but "acetylsalicylic acid" and all of its known brand names, as well as generic words and brand names of conceptually similar drugs—other painkillers in the same family as aspirin. The same would be done for "cure," bringing up search results for other similar words such as "treat" and "relieve," and for "headache," looking up results for specific types of headaches (child concepts of HEADACHE), as well as other similar painful conditions (parent concepts of HEADACHE, or PAIN in the same area).

Thus, an OntoSem competent search engine works in synergy with the user's knowledge: if the user specifically wants to know if aspirin treats tension headaches, the search will return results only for tension headache (which has no child concepts) and its parent concept, HEADACHE. If the user is less clear on what kind of information they want, and search only for "headache," the results that are returned will be more widespread, including general aches and pains centered around the head, neck, and shoulders.

Here, the implementation to web search determines the design of the OntoSem resources ontology and lexicon: How many concepts should be created under PAIN and CURE, how to distribute meaning between the ontological concept and the lexicon sense for the task at hand as well as future use.

This becomes particularly important for parallelizing queries in the way described above. Parallelization is defined as taking all similarly relevant words for a search query given an input word. For the example, the goal is to take a word-concept pair like (TREAT, TREAT-ILLNESS) and get lexicon entries like "care for." This is currently achieved with the following simple algorithm:

- Take the (word, concept) pair for the word you want to parallelize.

- Return all words grounded in the concept.

This is sufficient assuming that the lexicon and ontology are of a certain quality, namely, lexical entries that are mapped onto the same concept are indeed sufficiently "parallel," "similar," of the right degree of "synonymy." Here, again the implementation guides us to decide what "synonymy" means. Any card-carrying linguist will tell you that no two words in a language are fully synonymous, as that would be a frivolity that Language, ever economical, cannot afford. On the other hand, for internet search, "synonymy" can be operationalized as the relation between two lexicon entries a and b, where entry b should also be considered a correct search result, when entry a is found in the search query. In domains where the ontology or lexicon need improvement to conform to this requirement, parallelization can yield wild results and requires further support by adapting the algorithm.

With our current resources, parallelization from "aspirin" will yield all drugs mapped onto the concept DRUG, because the current algorithm only looks at heads of sem-strucs, the part of a lexicon entry that uses ontological concepts to specify the meaning of the sense (see below), which are DRUG for all of them. If we create a daughter concept PAINKILLER for "aspirin" and other painkillers the current algorithm won't parallelize any more as "aspirin" and "Preparation H" are grounded in different concepts. Let's assume, we don't want the parallelization from "aspirin" to "Preparation H," we can (a) improve the algorithm to take full sem-strucs into account or only consider fully identical sem-strucs, or improve the resources to scenario.

For resource improvement, based on the implementation purpose, we generated data by looking at a frequency list, picking all senses that are rooted under the event branch of the ontology, and then outputting the head concepts for those senses in the order found in the frequency list. Some concepts only have 10-20 other words rooted in them, at least one has 665, many have around 100.

We don't want the lexicon and ontology to approach each other in terms of entries. However, for some of these higher level concepts, the quantity of senses that are attached to a concept warrants the generation of more nodes beneath it for the purpose of the parallelization.

In addition, we are currently making the algorithm more discerning by enabling it to read sem-strucs in more detail, e.g., allowing for identical constraints in the agent/theme/instrument case roles to be criteria for parallelization.

5.2. Matching Queries to Pages through Meaning

Offline, robust, full OntoParsing text on crawled webpages into its TMRs is the largest-scale role of OntoSem in Internet search. As a simplified overview, the process includes

- web crawling,
- OntoParsing to produce the TMR of all clauses,
- summarization of webpage for topic identification and marking the degree of relevance of a sentence with respect to a page's content,
- TMR decomposition,
- breeding of anticipated queries to which the sentence is a relevant answer, and
- storage for fast retrieval of the anticipated queries in a non-index data structure, an innovative approach that allows the accommodation of processing- and storage-intensive meaning-based search technology.

In OntoParsing, no loss with respect to the input sentence is allowed. The common solution for this in NLP is to consider unattested input to be a named entity. With OntoSem we can go further and infer its general concept class, based on identifying its semantic role in the event described, and to pass it on into the onomasticon as a name for later use. Our onomasticon is intended to cover a large amount of named entities, as they are particularly relevant for Internet search, where queries frequently contain names of products and people.

Sentence (3) is OntoParsed into the TMR in (4), where we omitted the nested instrument relation of the first event of concept PROVE ("establish") and show only the theme of this event, the event SHOOT and its properties.

(3) [The FBI report through scientific examination of evidence, testimony and intensive investigation, established beyond a reasonable doubt that] Lee Harvey Oswald shot President Kennedy on November 22, 1963. (http://www.acorn.net/jfkplace/09/fp.back_issues/27th_Issue/vs_text.html)

(4)

shoot
- agent: human-1
  - has-name "Lee Harvey Oswald"
- beneficiary: human-2
  - has-name "President [John] [F.] Kennedy"
- time: 1963-11-22 ??:??:??

To be able to match the TMR in (4) to typical queries, such as (5) and (6), in particular, to do so fast, the following processing steps are necessary.

(5) who shot jfk

(6) kennedy murder

Queries are fast-OntoParsed. First, an event is identified, here SHOOT from "shot" (5) and KILL from "murder" (6). If not otherwise specified syntactically (trough word order, prepositions, etc.)—which is typical for search queries—the other concepts fill the properties of this event constrained only by semantic constraints of the ontology. "who," "jfk," and "kennedy" as HUMAN are mapped onto both AGENT and BENEFICIARY, the only possible case-roles for HUMAN. "date" is mapped onto TIME. The following are the resulting fast-TMRs for (5) and (6):

(7)

shoot
- agent: *human-1
- beneficiary: human-2
  - has-name "[President] J[ohn] F.]Kennedy"

(8)

shoot
- agent: human-1
  - has-name "President [John] [F.] Kennedy"
- beneficiary: *human-2

Note the asterisk that specifies the concept the query is asking for. Since there is no marking for passive in (5) and (6) is actually a syntactically correct query, the word order forces the (8) as the preferred TMR.

TMR (4) matches query (7) and the webpage from which the sentence OntoParsed as (4) was taken (and that is relevant, in that it has the shooting of Kennedy as its topic as witnessed from our OntoSem summarization) is retrieved as the answer.

A note on disambiguation is due: "jfk" (5) as well as "kennedy" (6) need to be identified as the same onomasticon entry as human-2. In the case of "jfk" there is no ambiguity for our system, as this is simply a synonym entry in our onomasticon, alongside many others to refer do kennedy-n2 ("John Fitzgerald Kennedy"). Handling "kennedy" is possible

through our multi-word expression storage format. The inherent ambiguity of "kennedy" as anyone with the name part "kennedy" is reduced through our selection of only 13 Kennedys for our onomasticon, mostly politicians from the same family, but also authors, actors and musicians. From the fact repository containing TMRs of previously OntoParsed webpages, we know that only kennedy-n2 ("John Fitzgerald Kennedy") and kennedy-n6 ("Robert Francis Kennedy") are beneficiaries of SHOOT events. Thus, (11) remains ambiguous, but now only two-way ambiguous, and any TMR based on the event SHOOT and with the beneficiary kennedy-n2 or kennedy-n6 would be a matching answer.

Constraints on ontological concepts can also help identify concept types of unattested input to automatically acquire it into the onomasticon.

(9) Selma Blair starred in the movie The Sweetest Thing alongside Cameron Diaz.
(www.filmspot.com/people/7528/selma-blair/trivia.html)

While we have an entry for Cameron Diaz as ACTOR-DRAMATIC, we don't have one for Selma Blair. Apart from Tony Blair as PRIME-MINISTER, only Linda, Janet, and Betsy Blair are covered in the onomasticon as ACTOR-DRAMATIC. But since the only verbal sense of "star" is PERFORM-CHARACTER, which is constrained for agent by default as ACTOR-DRAMATIC, we can add an entry to the onomasticon for Selma Blair. Such automatically acquired entries are flagged for human approval before they're fully incorporated. Also, often the constraint is not as strict, so that a named entity may only be mapped onto a as a generic concept, like CORPORATION or HUMAN, and will then have to be further specified during human quality control.

(10) There's something Eugene Mirman said in a monologue somewhere, which I'll now paraphrase to test out the "block quotes" feature.
(http://terribleposture.blogspot.com/)

In this example, the system knows that Eugene Mirman is HUMAN, since "say" is mapped onto the concept ASSERTIVE-ACT, which inherits from its parent ILLOCUTIONARY-ACT the constraint, that its agent must be HUMAN. If our system hits sufficient instances of Eugene Mirman either in the queries or in the crawled pages, it will increase the priority of quality controlling the automatically acquired onomasticon entry for that named entity and it can be further specified, e.g., as ENTERTAINMENT-ROLE.

6. Conclusion

Besides developing what is hoped to be a successful next-generation Internet search which will raise the users' expectations significantly beyond popularity algorithms (Brin and Page 1998) , the ontological semantic approach to NLP brings forth an essentially new discipline of Meaning Processing, as opposed to the BOW-and-statistics NLP. In the latter area, dominated for a variety of academic and sociological reasons, by non-linguists and non-semanticists, the approach emphasizes the significance of the meaning resources underlying human understanding of language and the commitment to developing them. The current NLP approach, on the other hand, is still trying to get at the meaning without penetrating the semantic substance of language, while trying to use ready-made (and, rarely, to create) word and frequency lists, WordNets, OWL formalisms, and other resources that are simple to acquire and used for that very reason. From the point of view of ontological semantics, these attempts to get at the meaning without doing semantics look like the *perpetuum mobile* project, probably highly desirable but not realistic and, most certainly, not accurate enough to be acceptable to the human user. It must be noted, however, that this attitude is highly contaminated by our decisive commitment to the representationalist, AI-type position.

7. References

[1] Berners-Lee ,T., J. Hendler, and O. Lassila. 2001. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities." Scientific American 284: 34-43.

[2] Brin S. and L. Page. 1998. "The anatomy of a large-scale hypertextual Web search engine." Computer Networks and ISDN Systems 30:1-7: 107-117.

[3] Chomsky, N. 1965. Aspects of the Theory of Syntax. Cambridge, MA: MIT Press.

[4] Deerwester, S., Susan Dumais, G. W. Furnas, T. K. Landauer, R. Harshman. 1990. "Indexing by Latent Semantic Analysis". Journal of the American Society for Information Science 41 (6): 391-407.

[5] Fellbaum, C. 1998. WordNet. An Electronic Lexical Database. Cambridge, MA: MIT Press.

[6] Hempelmann, C. F. 2007. "Beyond proof-of-concept: implementing ontological semantics as a commercial product." In: Raskin, Victor and John Spartz (eds.). Proceedings of the 4th Midwest Computational Linguistics Colloquium 2007. Purdue University, West Lafayette, Indiana. April 28, 2007.

[7] KBAE 2002. Knowledge-Based Acquisition Editor. Purdue Version 2.1. http: //kbae.cerias.purdue.edu:443/; guest login (browsing only), nlpgroup; password, ch@ng3me. NLP Lab and CERIAS, Purdue University, W. Lafayette, IN.

[8] Kelvin, W. Thomson, Baron. 1889. Popular Lectures and Addresses. London: Macmillan.

[9] Lyons, J. 1963. Structural Semantics: An Analysis of Part of the Vocabulary of Plato. Oxford: Blackwell.

[10] Nietzsche, F. 1901. Der Wille zur Macht. Leipzig: Naumann.

[11] Nirenburg, S., and V. Raskin 2004. Ontological Semantics. Cambridge, MA: MIT Press. A Pre-publication draft, as of the Fall of 2001, is available at http://crl.nmsu.edu/Staff.pages/Technical/sergei/book/index-book-html).

[12] Nirenburg, S., V. Raskin, and A. Tucker. 1987. "The Structure of Interlingua in TRANSLATOR." In: Nirenburg, S. (ed.) Machine Translation: Theoretical and Methodological Issues, NY: Cambridge University Press: 90-113.

[13] Obrst, L. 2007. "Ontology and ontologies: why it and they matter to the intelligence community." Proceedings of the Second International Ontology for the Intelligence Community Conference. OIC-2007. Columbia, MD. November 28-29.

[14] Raskin, V. 1971. K teorii yazykovykh podsistem /Towards a Theory of Language Subsystems/. Moscow: Moscow State University Press.

[15] Raskin, V. 1986. "Script-Based Semantic Theory." In: Ellis, D. G. and W. A. Donahue (eds.), Contemporary Issues in Language and Discourse Processes, Hillsdale, NJ: Erlbaum. 23-61.

[16] Raskin, V., C. F. Hempelmann and K. E. Triezenberg. 2008. "Ontological semantic forensics: Meaning-based deception detection." Paper Submitted to the 23rd International Information Security Conference (SEC 2008). Milan, Italy - September 8-10, 2008.