# Three Experiments on Mining the Web for Ontology and Lexicon Learning

Sergei Nirenburg sergei@umbc.edu
Donald Dimitroff dondim1@umbc.edu
Jesse English english1@umbc.edu
Craig Pfeifer cpfeifer@acm.org

Institute for Language and Information Technologies
University of Maryland, Baltimore County
Baltimore, MD 21250, USA

## ABSTRACT

This paper describes an approach to alleviating the well-known problem of the knowledge acquisition bottleneck in knowledge-based systems. In knowledge-based, meaning-oriented natural language processing, the core knowledge resources are a semantic lexicon and an ontological world model in terms of which lexical meaning is expressed. We describe a mutual bootstrapping approach whereby existing resources are used to create additional resources that are then added to the original resources. Thus, text understanding bootstraps the learning process , which in turn boosts the knowledge resources underlying text understanding. Specifically, in our experiments an existing ontology and an existing semantic lexicon are used by the ontological-semantic text analyzer OntoSem to analyze sentences mined from the web and containing specific words unknown to the system to generate candidates for ontological concepts and lexicon entries that at the time are not part of OntoSem's static knowledge resources. The experiments described in the paper have as their goal a) empirical determination of the number of senses for a word; b) automatic creation of ontological concepts (named sets of property-value pairs) describing the meanings of word senses; and c) suggesting the location of the newly acquired concepts in the ontological network. The experimental environment described is also used for empirical validation of ontological property values in concepts originally encoded by knowledge engineers.

## 1. INTRODUCTION

Automating knowledge acquisition is perhaps the single most important long-term goal of the field of intelligent systems and AI. Of all the possible sources of knowledge, text is probably the most attractive. A variety of methodologies have been employed to tackle the problem of automatic acquisition of knowledge from text, for example, statistical methods in conjunction with part of speech tagging or semantic clustering of known and unknown words [18], or generic pattern extraction for determining semantic relations [28]. We focus our knowledge acquisition approach on automatic extraction of meaning from texts using the OntoSem text analyzer and its associated knowledge resources (see Section 2). Our approach uses the Web as the open corpus of English texts to be processed by OntoSem. The knowledge structures obtained through OntoSem processing provide the basis for extending the coverage and improving the quality of OntoSem's knowledge resources, primarily, its ontology and lexicon. Thus, this approach is mutually bootstrapping, as the existing resources are used to support a process that results in their own expansion and enhancement. This approach can be seen as following two of the trends that Manning [20] described as essential for continued progress in machine learning of natural language – reliance on representations and on deeper interest in the features used for learning: "What … determines the better systems? The features that they use… This viewpoint is still somewhat unfashionable, but I think it will increasingly be seen to be correct… The often substantial difference between the systems is mainly in the features employed. In the context of language, doing "feature engineering" is otherwise known as doing linguistics. A distinctive aspect of language processing problem is that the space of interesting and useful features that one can extract is usually effectively unbounded. All one needs is enough linguistic insight and time to build those features (and enough data to estimate them effectively)." Our work certainly relies on representations and also on a set of ontological features that were developed and tested in various semantic processing engines over many years.

In this paper we present our latest results in ontology and lexicon learning by mining the Web. Ontology learning as a field concerns itself at this time with learning terms, (multilingual) synonyms, concepts, taxonomies (by far the most popular topic), relations and rules and axioms [6]. Different combinations of linguistic (knowledge-based) and statistical methods are typically used, but mostly the latter. Work on extracting specific relations using largely statistical means has been reported – [7] for meronymy, [9] for the qualia of the generative lexicon approach [30], and causal relations [16], among others. OntoSem, however, addresses the task of extracting knowledge about a large set of such relations using encoded knowledge as heuristics. Thus, our goals are closer, for example, to work by [10] that uses essentially statistical methods for estimating selectional restrictions. Among the sources of knowledge acquisition are machine-readable dictionaries (e.g., [24]), thesauri (e.g., [23]), as well as text (e.g., [26], [5], [8]).



*KDD07*, August 12–15, 2007, San Jose, California, USA.

Our approach relies on a dynamically generated corpus of knowledge structures, text meaning representations, or TMRs generated by OntoSem (see Section 2), which relies on deep linguistic analysis strengthened by statistical algorithms operating over an ontology and the nascent TMRs. At present, the quality of automatically generated TMRs is not optimal. A long-term goal of our work is to improve the quality of TMRs through learning new ontological and lexical knowledge using the current state of OntoSem, with or without using human validators/editors to "goldenize" system-produced TMRs.

This paper is organized as follows. Section 2 briefly describes the ontological-semantic environment that we use for mining data from the Web. The experiments and results are described in Section 3. Integration of separate results and evaluation is presented in Section 4. Section 5 is devoted to discussion of the results and future work.

## 2. ONTOSEM

OntoSem (the implementation of the theory of Ontological Semantics; [25]) is a text-processing environment that takes as input unrestricted raw text and carries out preprocessing, morphological analysis, syntactic analysis, and semantic analysis, with the results of semantic analysis represented as formal text-meaning representations (TMRs) that can then be used as the basis for many applications. TMRs have been used as the substrate for question-answering (e.g., [2]), machine translation (e.g., [1]) and knowledge extraction, and were also used as the basis for reasoning in the question-answering system AQUA, where they supplied knowledge to showcase temporal reasoning capabilities of JTP [15]. Text analysis relies on the following static knowledge resources:

- The OntoSem language-independent **ontology**, which currently contains around 8,500 concepts, each of which is described by an average of 16 properties. The ontology is populated by concepts that we expect to be relevant cross-linguistically. The current experiment was run on a subset of the ontology containing about 6,000 concepts.
- An OntoSem **lexicon** whose entries contain syntactic and semantic information (linked through variables) as well as calls for procedural semantic routines when necessary. The current English lexicon contains approximately 30,000 senses, including most closed-class items and many of the most frequent and polysemous verbs, as selected through corpus analysis. The base lexicon is expanded at runtime using an inventory of lexical (e.g., derivational-morphological) rules.
- An **onomasticon**, or lexicon of proper names, which contains approximately 350,000 entries.
- A **fact repository**, which contains "remembered instances" of ontological concepts. The fact repository is not used in the current experiment but will provide valuable semantically-annotated context information for future experiments.
- The OntoSem syntactic-semantic **analyzer**, which performs preprocessing (tokenization, named-entity and acronym recognition, etc.), morphological, syntactic and semantic analysis, and the creation of TMRs.
- The TMR language, which is the **metalanguage** for representing text meaning (a converter was developed between this custom language and OWL, see [17]).

OntoSem knowledge resources have been acquired by trained acquirers using a broad variety of efficiency-enhancing tools – graphical editors, enhanced search facilities, capabilities of automatically acquiring knowledge for classes of entities on the basis of manually acquired knowledge for a single representative of the class, etc.

## 3. THE EXPERIMENTS

Our research aims at automatic enhancement of both the ontology and the ontological-semantic lexicon. One goal is to mine the Web to learn the meanings of words unknown to OntoSem. Another goal is to mine the Web to provide empirical verification for the values of the various ontological properties that were acquired by human knowledge engineers. We make a simplifying assumption that the meaning of a word unknown to the system will be expressed as a univocal mapping to an ontological concept. This decision does not constrain the results, though it influences the interrelationship between the ontology and the lexicons in OntoSem – under the current assumption, the meaning of a word, as recorded in the *sem-struc* zone of its lexicon entry, will be simply a pointer to an ontological concept. This univocal mapping is just one of several types of lexical meaning specification in OntoSem (see [25], Chapter 8 for details).

In order to learn an ontological concept by mining the Web to establish the meaning of a(n unknown) word, one must a) determine a set of ontological properties relevant to the newly acquired concept; b) determine the ranges of values of these properties; and c) find an appropriate place in the ontological hierarchy to add the new concept. Unknown words can be polysemous, in which case it would also be necessary to d) determine the appropriate number of senses for the unknown word and create a new ontological concept for each of them. Determining the number of senses is a difficult task in itself (practically no two dictionaries have the same number of senses for a word). Note that whatever process is used for determining the number of senses for an unknown word can also be applied to words already in the lexicon, as it is quite possible that in the existing lexicon not all of whose senses are covered.

The three experiments we are reporting are devoted to the specific subtasks in the list above. Some of the processing is identical in each of the experiments. But they differ in the degree to which they use the various OntoSem resources. Thus, Experiment A is devoted to empirical validation of existing property (specifically, attribute) values and learning new ones. It uses only the static knowledge resources of OntoSem. Experiment B seeks to determine the number of senses of the new words and delineate the constraints on the property (mostly, relation) values of each sense. This experiment, in addition to the static knowledge resources, also relies on the results of syntactic analysis of input. The main goal of Experiment C is to determine the semantic constraints on the property values of the newly acquired concepts. Finding the appropriate positions for the new concepts in the ontological hierarchy is an auxiliary objective that facilitates the evaluation of the results. We also experiment with integrating all of our experimental work. In particular, we report on how results of Experiments A and B can be used as additional empirical evidence in Experiment C.

The method of Experiment A is "knowledge-lean" while that of Experiment C is "knowledge-rich," with Experiment B somewhere in between on this scale. Knowledge richness promises better results due to availability of human-acquired prior descriptive and processing-oriented knowledge that plays the role equivalent to that of corpus annotation in methods that are more

clearly statistical. It is common knowledge, however, that creating knowledge manually is expensive and prone to errors – after all, this difficulty is widely considered the main reason for the lack of success of early AI. Many recent and current approaches to both learning from text and processing text have therefore tended toward knowledge-lean methods. This preference promotes broad coverage at the expense of the quality of results. The difference in the amounts of knowledge and processing resources in the three experiments reported in this paper was intended to provide empirical data on the utility of knowledge-lean data mining for eliciting rich ontologies and suggest possible economies in the acquisition effort – in part, by allowing human acquirers to be instead validators of knowledge generated automatically.

All the experiments share some of their steps. Each experiment starts with a list of words whose meanings will be learned by the system. Next, each experiment mines from the Web a corpus of sentences containing this word. In the case of Experiment A, a sequence of search queries is created, in which the word in question is augmented by the English realizations of the meanings of ontological attributes. Experiment A concentrates on attributes (unary predicates with either numerical or symbolic ranges as value sets). Experiment B concentrates on case roles (a subset of ontological relations that includes such relations as agent, theme and beneficiary, among others) that reflect the constraints on co-occurrence of various ontological concepts in propositions encoding meanings of natural language sentences. This experiment uses the syntax-semantics linking information in the OntoSem lexicon to acquire sets of semantic constraints on the basis of the meaning sets corresponding to specific syntactic roles in phrases with the unknown word. For example, one such set would contain ontological concepts that specify the meaning of all the attested words that serve as syntactic heads of phrases serving the grammatical function of DirectObject of a verb. Determining the number and grain size of concept clusters in the above sets of semantic constraints is the main means of suggesting the number of senses for the unknown word.

Experiment C uses both syntactic and semantic analyzers of OntoSem seeking to produce TMRs for sentences containing the unknown word. The OntoSem text analyzer degrades gracefully in the face of unexpected input, so it is capable of semantically analyzing sentences with a small number of unknown words by assuming that the unknown word's meaning corresponds directly to a non-existent ontological concept and then (unidirectionally) applying relevant constraints listed in the ontological interpretations of the meanings of the known words in input that are connected with the unknown word through well-defined ontological relations to hypothesize the constraints on the meaning of the unknown words. As a result of this stage, Experiment C produces sets of pairs of property instances and their values that are, in effect, the newly acquired ontological concepts. In many cases OntoSem is not capable of carrying out unidirectional selectional restriction matching, so that not all the sentences containing the candidate word that are found in the corpus yield useful property-value pairs. Once the set of such pairs is found, the system compares it to other such sets that comprise the ontological descriptions of concepts already existing in the OntoSem ontology, thus suggesting a place for the new concept(s) in the ontological hierarchy.

## 3.1 Experiment A: Mining the Web for Attribute Values

In this experiment the OntoSem ontology is used as the basis for building search queries. Each ontological property of the attribute (unary) type is associated with a list of its possible English realizations (obtained from the system's lexicon). A search query is created by combining this list with either a word (when learning new concepts and lexicon entries) or a list of words realizing in English the meaning of a concept (when using this method for empirical verification of the existing ontology and lexicon). For example, for the concept ELEPHANT and the attribute WEIGHT, the following query is produced: (elephant) AND (weigh OR mass OR heavy OR heaviness). Note that since Google matches partial strings on queries, the search string *weigh* will match with many strings such as: *weigh*, *weight*, *weighing*, *weighs*, *weighed*, etc. Typically, a search of 500 web pages takes approximately one hour.

The result of the search is a list of sentences matching the query. These candidate sentences are then processed further by one of two different methods depending upon whether they contain a measurable (e.g., weight) or non-measurable (e.g., color, whose values are represented in the OntoSem ontology by a set of primitive literals, such as *green*) attributes. In the latter case, a count is produced for the occurrences of each of the literals in the data. Table 1 illustrates the results for several runs aimed at empirical validation of existing ontological values. These searches were done on 500 web pages.

**Table 1. Learning Literal Attribute Values**

| Concept/ Attribute | Web Mining Results | Existing ontology has |
|---|---|---|
| ELEPHANT/ COLOR | white: 283; pink: 188; blue: 92; black: 64; red: 61; gray: 45; green: 39; yellow: 39; brown: 23; purple: 13 | black<br>brown<br>gray<br>tan<br>white |
| SPINACH/ COLOR | red: 509 green: 297<br>black: 246 white: 227<br>orange: 129 blue:125 | green |
| GRATER/ SHAPE | conical: 33; circular: 22; curved: 27; cylindrical: 23; rectangular: 23; hexagonal: 12 | parallelepiped<br>sheetlike<br>trapezoidal |

Table 1 suggests that the results will have to be validated by humans before they are actually used to modify the ontology due to empirical evidence. The great variety of elephant colors appears generally because of metaphorical usage (pink, white) or because the elephants are toys. While there are indeed both green and red varieties of spinach, the other colors are actually of other objects in the search space returned by the queries. At the same time, the last line of the table does not contain the same percentage of noise and is rather useful as is.

In the case of scalar attributes, each sentence in the results is searched for value-unit pairs. For example, the sentence "*The elephant weighs five tons and is ten feet tall*" contains two such pairs: *five tons* and *ten feet*. If the property being searched for is WEIGHT, then five tons is accepted a valid measure, since tons is

a weight measure, while ten feet is rejected since weight is not measured in feet. All units are then converted into metric units, which is the standard in the OntoSem ontology. In some special cases, stop lists were generated to eliminate any errors that might be introduced by such conversions. Care was also taken to find ranges of values. If a sentence said *"Elephants weigh between 4000 and 9000 kilograms."* then both 4000 kilograms and 9000 kilograms are returned as valid elephant weight values. If the purpose of the data mining run is to determine constraints on a property of a newly learned concept, the range of values mined from the Web is compared with the range of values for the attribute in question within the definition of the concept in question in the OntoSem ontology. Table 2 presents a small sampling of results based on a search of 200 web pages for each concept-property pair.

**Table 2. Concept-Property Pairs for Scalar Attributes**

| Concept | Attribute | Range Mined from the Web | Range in Ontology |
|---|---|---|---|
| SQUASH | LENGTH | 0.025 – 9.754 (meters) | 0.012 – 0.024 (meters) |
| TUNA | WEIGHT | 0.128 – 817 (kilograms) | 2 – 820 (kilograms) |
| ELEPHANT | WEIGHT | 0.227 – 10866 (kilograms) | 3500 – 13000 kilograms |

The above results are still noisy due to a variety of reasons but mainly, again, because of ambiguity (toy elephants versus real elephants). A central direction of our work is studying the cost of eliminating this ambiguity against the depth and breadth of knowledge that must be used for disambiguation. The first preliminary step in that direction is determining the cardinality of the set of concepts that can correspond to the word in question, that is, the arity of the ambiguity. For example, elephants can be ambiguous between African, Asian, forest, toy and metaphorical elephants. Our initial approach is that of clustering the results obtained from mining the Web. We clustered the results of Experiment A using the EM algorithm tool in the WEKA toolkit [32]. As an example, consider the results of clustering for SQUASH LENGTH. Four clusters were produced. The fourth cluster, which ranged from 4.877 to 9.754 meters, included values that were all derived from sentences which dealt with *squash courts*. This kind of outcome again demonstrates a limitation of the knowledge-lean approach used in this experiment. For better results, one must analyze the data deeper; in this case, it will help to filter out the cases where the word *squash* is not the head of the noun phrase in which it appears. However, if used for empirical support of a human knowledge acquirer, even the current clustering capabilities facilitate knowledge acquisition – for instance, the elimination of the cluster relating to squash courts from consideration of the length attribute of squash the vegetable (see Table 3). The results for TUNA and ELEPHANT WEIGHT in this table similarly filter out the weight of cans of processed tuna, baby elephants, elephant parts (brain, heart, tail, etc.) and elephant seals.

**Table 3. Concept-Property Pairs for Scalar Attributes after filtering by human acquirers**

| Concept | Attribute | Range Mined from the Web | Range in Ontology |
|---|---|---|---|
| SQUASH | LENGTH | 0.025 – 0.914 (meters) | 0.012 – 0.024 (meters) |
| TUNA | WEIGHT | 2.5 – 817 (kilograms) | 2 – 820 (kilograms) |
| ELEPHANT | WEIGHT | 2268 – 10866 (kilograms) | 3500 – 13000 kilograms |

## 3.2 Experiment B: Word Sense Discrimination and Delineating Relations Among Verb Senses

This experiment uses one of OntoSem's static knowledge sources, the lexicon, in conjunction with syntactic analysis and unsupervised clustering to automatically determine the number of senses detected in a corpus for a an unknown verb and generate for it a candidate OntoSem lexicon entry. For our initial experiment, each automatically generated cluster of values of the agent and theme case roles will represent a unique sense of the unknown verb. Once the candidate lexicon entries are created, they can be presented to knowledge acquirers for validation. Results of this experiment are also integrated with those of Experiment C (see Section 4 below).

The process used is as follows. The web is searched for documents that contain any form of an unknown verb. The documents are prepared for processing by stripping any HTML markup and chunking the remaining content into sentences using LingPipe [19]. These sentences are then searched for any morphological form of the unknown verb. The matching sentences are parsed using the Bikel parser [4]. Syntactic analysis yields immediate constituents for a sentence. A set of simple heuristics establishes which of the constituents fill the grammatical functions of subject and direct object. This subcategorization information, in turn, facilitates the use of the OntoSem lexicon to suggest a set of semantic constraints on the AGENT and THEME case roles of the concept underlying the unknown verb. For simplicity, we assume only a single type of linking between grammatical functions and semantic selectional restrictions (case roles): the grammatical subject is linked to the AGENT case role, while the grammatical object is linked to the THEME case role.

Similar approaches to the task of word sense discrimination include [33], [31], and [29]. [33] uses seed rules to create training examples for a supervised classification algorithm. These seed rules define word cooccurence rules over a span of tokens and do not use any syntactic or semantic knowledge from the text. [31] also uses word cooccurence, however second order cooccurences are used to reduce sparsity. [29] uses word collocations and coocurrences as features for unsupervised clustering, but include linguistic features such as part of speech tags and content collocations. Our approach is more similar to [29], in that we use a mix of word and syntactic features for unsupervised clustering. However, the inventory of linguistic and especially semantic features we are able to use is more extensive, because we can take advantage of the Ontosem lexicon.

A look-up in the OntoSem lexicon attaches one or more ontological concepts to words in the context. Because, unlike in Experiment C, compositional-semantic analysis is not performed, lexical ambiguity cannot be eliminated, and as a result, several ontological concepts can in principle be attached to a word. For this experiment we make a strong simplification by choosing only the concept from the first sense of a word in the OntoSem lexicon.

We use a simplifying hypothesis that the head of the noun phrase immediately prior to the verb is most likely to be the subject of the verb and the head of the noun phrase following the verb is most likely to be the direct object of the verb. These features give information about the syntactic and token context of the verb. However to identify distinct word senses, we add the semantic features *agent* and *theme*. *Their values are* obtained from the OntoSem lexicon entries for the heads of the subject and direct object grammatical functions.

Once the features are determined, we generate a single feature vector for each sentence in our corpus. To compute the clusters over the features we use the Expectation Maximization (EM) algorithm [12] in the WEKA toolkit. Each cluster suggests a unique candidate word sense. The top 5 values for the features *agent* and *theme* are used to determine constraints on the THEME and AGENT semantic roles of the candidate lexicon entries. The top 5 values for the *subject* and *object* features give the human reviewer the insight of what has generated the top 5 concepts.

### *3.2.1 Experimental Results*

Tables 4 and 5 show the top 5 words and concepts for the two clusters generated by data collected for the unknown verb *deport*. In the OntoSem lexicon, *deport* is currently monosemous. Even though deport currently exists in the lexicon it is not used for parsing or other syntactic tasks. This does not skew the results, and it gives an initial metric to compare our results against.

**Table 4. 5 Most Frequent Values for Verb Deport, Cluster 1**

| Subject | Agent | Object | Theme |
|---|---|---|---|
| Government | Nation(48) | people | Human (144) |
| ICE | Human (41) | Aliens | Nation (31) |
| Policy | Procedure (13) | Country | City (27) |
| officials | Ice (13) | Germany | Citizen (14) |
| Authorities | Social-role (10) | Immigrants | Criminal (14) |

**Table 5. 5 Most Frequent Values for Verb Deport, Cluster 2**

| Subject | Agent | Object | Theme |
|---|---|---|---|
| Jews | Human (145) | Auschwitz | Nation (110) |
| People | City (31) | Deportation | Human (52) |
| Immigrants | Nation (21) | Country | City (23) |
| Persons | Function (10) | Time | Printed-media (8) |
| Person | Social-role (8) | People | Year (8) |

Tables 4 and 5 show the top 5 values by frequency of the clustered instances for the four features *subject*, *agent*, *object* and *theme*. The frequency counts for each value in the corpus are shown in parentheses. Note that there is no special significance to the fact that, say, the fifth ranked object word is *immigrants* and the fifth ranked object concept is CRIMINAL. If a context word does not have a corresponding concept (that is, it is not in the OntoSem lexicon) or if the unknown verb is intransitive, values for the *object* and *theme* features would be missing. Based on the data in Tables 4 and 5, two candidate lexicon entries would be produced for human review*:*

```
(deport-v1
  (cat                                                    v)
  (syn-struc ((root $var0) (cat v) (subject ((root $var2) (cat np)))
   (directobject ((root $var3) (cat np)))))
 (sem-struc
   (event (agent (value NATION)) (theme (value HUMAN))))))

(deport-v2
  (cat v)
  (syn-struc ((root $var0) (cat v) (subject ((root $var2) (cat np)))
   (directobject ((root $var3) (cat np)))))
 (sem-struc
   (event (agent (value HUMAN)) (theme (value NATION))))))
```

Note that the position of Experiment B on the knowledge richness scale is such that it does not take into account the differences in voice (active or passive) or other diathesis transformations of the sentences mined from the web. In the above example, it is for this reason that two lexicon entries are automatically suggested.

In our example, both candidate lexicon entries are transitive verbs. One candidate entry will have its AGENT and THEME case roles constrained to the concepts in the ontological subtrees rooted at the concepts NATION and HUMAN, respectively. The second sense will have this assignment reversed. Depending on the specific task at hand, the knowledge acquirer may decide to include either both senses, or any one of them.

The second ranked subject word in cluster 1, ICE, is not frozen water but rather an acronym for Immigrations and Customs Enforcement. A similar error occurs with the fourth ranked object word in cluster 2, *Time*, which refers to *Time Magazine*. The inclusion of a named entity tagger will help to resolve these errors**.**

### *3.2.2 Future Work*

Our current experiment is rather constrained in both the inventory of features used and the depth of analysis of text. We plan to extend the inventory of grammatical functions and, consequently, case roles used in the process by extending the set of verb subcategorization patterns covered to include prepositional objects (and their semantic correlates in the realm of case roles – e.g., DESTINATION or PURPOSE) by processing prepositional phrases attached to the unknown verb.

Intelligent feature engineering could be applied to generate nominal values for processing events such as “context word not in lexicon,” “no ontological concept for context word,” and “fewer than 10 context words exist” as opposed to simply marking these values as “missing”.

## 3.3 Experiment C: Learning Word Meanings (Ontological Concepts)

In this section, we describe our experiment devoted to learning meanings of words not covered by the existing OntoSem lexicon. We are making a simplifying assumption that the meaning of a word is represented through a univocal mapping to an ontological concept. To evaluate the quality of the machine-learned ontological concepts, we have also developed a procedure for finding the most appropriate place for the newly learned concept in the existing ontology. We can then compare the automatically derived ontological position with the best decision by human acquirers.

We start with an unknown word and mine the web for sentences containing it. These sentences are then processed using the OntoSem text analyzer. Because OntoSem has been engineered to handle unexpected input, it will not fail on a sentence just because it contains an unknown word. In the resulting TMR, the meaning of the unknown word W will be represented by an ontological concept C that will be created by unidirectional application of selectional restrictions listed with the meanings of those words in the input that stand in specific semantic relations with W. For example, if *cook* in its main verbal sense were an unknown word in the OntoSem lexicon, analysis of a large corpus of sentences with this verb would result in proposing to explain its meaning using a concept of the type EVENT, with constraints on its THEME deriving from the ontological subtrees rooted at the concepts PREPARED-FOOD and MEAL. The following description of the learning algorithm contains references to experimental results on twelve learned concepts, shown in Tables 6 and 7 of Section 4.

After OntoSem produces a set of TMRs (see Table 6, column B), the automatically generated property-value pairs for the candidate concept are extracted (see Table 6, column C). These properties are often relational properties, such as THEME, or AGENT-OF. These properties are then filtered down, in order to leave only the more relevant ones for further analysis. First, placeholder and debugging properties are removed. After this phase of processing, the results will be in the form of a list of property-value pairs similar to the following (numbers appended to the concept name are numbers of instances of that object remembered by the system):

<relation type="AGENT-OF" value="EVENT-2069">
<relation type="AGENT-OF" value="WEAR-CLOTHES-3642">
<relation type="BENEFICIARY-OF" value="EVENT-879">
<relation type="DESTINATION-OF" value="OBJECT-2932">

The next round of filtering eliminates those property-value pairs on the empirically generated list whose value sets are fully covered by the value set of another property-value pair for the same property. In other words, when there exist two instances of the same property, say, AGENT, with different values, say OBJECT, and PHYSICAL-OBJECT, the property instance with value OBJECT will be filtered out, because being a PHYSICAL-OBJECT presupposes being an OBJECT (in other words, PHYSICAL-OBJECT is an ontological descendant of OBJECT). As a result of this step, the first line of the above list will be deleted, as WEAR-CLOTHES subsumes EVENT.

The process of querying the web, analyzing the text, extracting and filtering the properties is repeated until an empirically defined minimum inventory of property-value pairs is collected. At this point, a candidate concept is declared to be the set of remaining property-value pairs.[1]

Now that the concept has been created, we start the process of finding an appropriate place for its inclusion in the ontological network. This process pursues two different goals. First, as mentioned above, it is used to evaluate the quality of the automatically generated concept candidate. Second, it serves as a means of integrating results of Experiment A with those of Experiment C. The reason for this integration is the realization that these experiments are, in a sense, complementary. Indeed, the method of Experiment C relies on automatic learning of relation-type properties, which is facilitated by the TMR-producing capabilities of the OntoSem analyzer (the reason being that English realizations of relations, such as AGENT or THEME co-occur with the unknown word whose meaning we are learning in any sentence with that word). For Experiment A to operate, it must know the inventory of relevant attribute-type properties to form appropriate queries for mining the web.

The process of finding the inclusion location for the candidate concept is described in detail in [14]. Here we present just a brief sketch. The candidate concept is compared in order with each concept in the ontology. An algorithm, a version of the SVM approach, for determining similarity on a 0 to 1 scale has been implemented for this purpose. The concept for which the algorithm returns the highest result is considered the "closest" match to the candidate concept, and marks the system's choice for where to insert the newly created concept into the ontological graph (see Table 6, Column E).

The similarity metric we use first identifies the type of the values for each property (numeric, numeric range, concept, concept range, literal, etc.) and then uses a set of specially designed heuristics for comparing each pair depending on its value type. Each value pair returns a similarity result, which we then use as a weight on the value of the matching property. Combining these weights we get a total similarity measure between any two concepts, without having to rely on the ontological hierarchy. At this time, we treat each property as equally important. It is clear that establishing a rating of property importance should enhance the quality of results. This task is an important item of future work in this project.

Overall, Experiment C uses a three-step procedure:

1. Learn candidate concept: a set S of property-value pairs
2. Derive ranked list L of positions for candidate in the ontology
3. Evaluate quality of the learning process

## 3.4 Results and Evaluation

To evaluate the quality of a newly learned candidate ontological concept, we automatically produce a ranked list of concepts that can serve as the candidate's parents or siblings in the ontological network and then compare elements of this list to the concept ("target concept") determined by a human judge to be the appropriate parent or sibling of the candidate. Bernstein [3] discusses two distinct methods of calculating similarity of concepts in an ontology: edge-based, and node-based (we will be

[1] At this time, we do not introduce a special naming procedure for the new concept candidate; in future, we plan to automate the naming policy that is followed by human acquirers of the OntoSem ontology.

using a combination of the two). Edge-based comparison has been implemented, for example, in the OntoSearch algorithm [27]. OntoSearch calculates a distance value between two concepts in a given ontology by traversing property paths, applying a weighted penalty to each crossed path.

Note, however, that in our case OntoSearch cannot be used initially as a basis of evaluation, as the candidate concept has – as yet – no place in the ontology, thus failing to meet one of the basic requirements for OntoSearch's usage. In order to identify a place in the ontology for the candidate, we carry out a pairwise comparison of all values defined in each property of the candidate, and all property-value pairs in each concept in the ontology; in other words we must do a node-based comparison.

Once the ranked list of potential attachment-point concepts for a candidate concept is produced, we can use OntoSearch to calculate the ontological distance between each member of this list and the target concept (see Table 6, column G). This distance is used as the measure of the quality of our method of ontological concept learning (and, consequently for our approach, learning meanings of words unknown to the system).

The results of a set of 12 runs of Experiment C are presented in Table 6. This table reflects the results of Experiment C only.

**Table 6: Results of Experiment C on Twelve Unknown Concepts**

| Word | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| Brontosaurus | DINOSAUR | 302 | 2150 | 0.373 | 0.492 | 9007 | 0.715 |
| Cherimoya | FRUIT-FOODSTUFF | 148 | 895 | 0.335 | 0.453 | 11546 | 0.637 |
| Deport | BANISH | 1043 | 4994 | 0.409 | 0.485 | 12503 | 0.679 |
| Depose | DEPOSE | 54 | 256 | 0.479 | 0.600 | 11079 | 0.999 (ALL) |
| Diplodocus | DINOSAUR | 469 | 2905 | 0.500 | 0.550 | 2290 | 0.546 |
| Obey | OBEY | 60 | 397 | 0.384 | 0.460 | 5370 | 0.518 |
| Pledge | PROMISE | 1323 | 5934 | 0.335 | 0.436 | 14097 | 0.760 |
| Spartan | MILITARY-ROLE | 426 | 2201 | 0.481 | 0.492 | 1409 | 0.754 |
| Stegosaurus | DINOSAUR | 415 | 3306 | 0.499 | 0.538 | 625 | 0.759 |
| Syrup | PLANT-DERIVED-FOODSTUFF | 322 | 1377 | 0.423 | 0.465 | 2315 | 0.760 |
| Triceratops | DINOSAUR | 84 | 796 | 0.482 | 0.488 | 588 | 0.849 |
| Wigger | SOCIAL-ROLE | 57 | 233 | 0.484 | 0.489 | 702 | 0.849 |

A: The targeted "correct" concept (existing in the ontology).

B: The number of clauses extracted containing the search word.

C: The total number of property/value pairs generated.

D: The calculated similarity between the candidate and target concept.

E: The calculated similarity between the candidate and the concept(s) with the highest similarity.

F: The rank of the target, out of approximately 16913, when compared to the candidate.

G: The best distance to the target, using relations only (by OntoSearch comparison standards).

Table 7 shows the combined results of Experiments A and C. In both cases, columns containing results from OntoSearch reflect the final similarity distance calculated between the candidate and target concepts. Table 8 suggests two alternate distance calculations, using OntoSearch, to those shown in Table 6; column B presents the results of the OntoSearch calculation when including all the attribute/value pairs generated in Experiment A, whereas column C presents the results when including only the attribute/value pairs who had the highest attributed incidence for that value.

# 4. Integrating the Three Experiments

Experiments A and B support Experiment C by providing, respectively an additional and alternative method for generating clustered attribute-value and relation-value pairs. Integrating this data into Experiment C is expected to enhance the learned knowledge of the candidate concept.

## 4.1. Adding Attributes to Experiment C

After performing the second step in Experiment C, a ranked list of suggested locations in the ontology where the candidate can be positioned becomes available. The system can now extract known attribute properties (WEIGHT, COLOR, etc.) of the ontological concepts that are selected as parents or siblings of the candidate concept and send them as input to the process of Experiment A. The experimental procedure, thus, is augmented as follows:

1. Learn candidate concept: a set S of property-value pairs
2. Derive ranked list L of positions for candidate in the ontology
3. Run Experiment A with elements of L as input
4. Augment set S using results of Step 3
5. Derive new list L using the augmented set S
6. Evaluate quality of the learning process

As an example, if the existing concept DINOSAUR was suggested by the original run of Experiment C as the parent of DIPLODOCUS, it is assumed that the definition of DIPLODOCUS will contain all the attribute-type properties that are defined for DINOSAUR (through inheritance). Therefore all these attributes will be used in a dedicated run of Experiment A. As a result, we enhance the inventory of the property values defining the candidate concept with attribute-value pairs of the kind:

<attribute type="HEIGHT" value="(<> 0.019 2.134)">
<attribute type="LENGTH" value="(<> 0.025 3.658)">

By appending the attribute-value pairs to the existing list L of relation-value pairs, we can proceed to the evaluation step with a more detailed candidate concept. Table 7 shows the results of integrating Experiment A with Experiment C as described above, the modified evaluation is shown in column D.

## 4.2 Adding Clustered Results of Experiment A to Experiment C

The integration method described in the previous section does not take advantage of the results of clustering obtainable in Experiment A. This information can be used as a filtering technique when appending

attribute-value pairs to the output of Experiment C to improve the quality of the candidate concept and to attempt to eliminate ambiguity on the basis of empirical data.

An output from Experiment A could include in the following four attribute-value pairs:

1. <attribute type="HEIGHT" value="(<> 0.019 2.134)">">
2. <attribute type="HEIGHT" value="(<> 2.743 3.048)">
3. <attribute type="LENGTH" value="(<> 0.025 3.658)">
4. <attribute type="LENGTH" value="(<> 4.572 12.000)">

Suppose (1) was extracted from a cluster of size 15, (2) from a cluster of size 16, (3) from a cluster of size 36, and (4) from a cluster of size 30. We can use this information to prune the data, instead of taking all accounts of the attributes (as suggested in the previous section).

Several methods can be used for suggesting which (if any) clusters should be used; one simple approach would be to select the value for the largest cluster within each unique attribute type, (this is similar to selecting the first sense of a word in the lexicon). Table 8 shows the results of such a selection, as contrasted with results from Table 7, and Table 6.

**Table 7: Results of Combined Experiments A and C on Eleven Unknown Concepts**

| Word | A | B | C | D |
|---|---|---|---|---|
| Brontosaurus | DINOSAUR | 15 | 0.607 | 0.715 |
| Cherimoya | FRUIT-FOODSTUFF | 17 | 0.607 | 0.637 |
| Depose | DEPOSE | 9 | 0.999 (ALL) | 0.999 (ALL) |
| Diplodocus | DINOSAUR | 13 | 0.612 | 0.546 |
| Obey | OBEY | 10 | 0.518 | 0.518 |
| Pledge | PROMISE | 14 | 0.516 | 0.760 |
| Spartan | MILITARY-ROLE | 17 | 0.574 | 0.754 |
| Stegosaurus | DINOSAUR | 21 | 0.720 | 0.759 |
| Syrup | PLANT-DERIVED-FOODSTUFF | 24 | 0.646 | 0.760 |
| Triceratops | DINOSAUR | 22 | 0.643 | 0.849 |
| Wigger | SOCIAL-ROLE | 8 | 0.526 | 0.849 |

A: The targeted "correct" concept (existing in ontology). Repeated for clarity.

B: The total number of attribute/value pairs generated.

C: The best distance to the target, using attribute and relations (by OntoSearch comparison standards).

D: The best distance to the target, using relations only (by OntoSearch comparison standards). Repeated for clarity.

## 4.3 Integrating Experiments B and C

Experiment B focuses on clustering the subjects and objects of verbs to produce statistical insight as to the number of unique senses of a verb. As Experiment B already uses the case roles AGENT and THEME that are among the properties processed in Experiment C, the integration of the two experiments is natural.

The experimental procedure is as follows:

1. Learn candidate concept: a set S of property-value pairs
2. Run Experiment B on the same input as Experiment C
3. Append values of AGENT and THEME from Experiment B to values obtained in Experiment C (adding together occurrence counts for values produced by both procedures)
4. Derive ranked list L of positions for candidate in the ontology
5. Evaluate quality of the learning process

Using the sample data shown Table 4, we can append the following property-value pairs to data already constructed by Experiment C, shown in Table 6, for the word *deport*:

1. <relation type="AGENT" value="NATION">
2. <relation type="AGENT" value="HUMAN">
3. <relation type="AGENT" value="PROCEDURE">
4. <relation type="AGENT" value="ICE">
5. <relation type="AGENT" value="SOCIAL-ROLE">
6. <relation type="THEME" value="HUMAN">
7. <relation type="THEME" value="NATION">
8. <relation type="THEME" value="CITY">
9. <relation type="THEME" value="CITIZEN">
10. <relation type="THEME" value="CRIMINAL">

The quality of the learning process on the basis of merged data from Experiments B and C increased from 0.679 (as reported in Table 6), to 0.721.

## 5. Discussion and Future Work

The work described in this paper is viewed as a step in a long-term program toward automating knowledge acquisition for meaning-oriented NLP applications. The overall methodology we are following is that of mutual bootstrapping – of the learning by the semantic text analysis capability and vice versa. We envisage a life-long learning environment in which a) newly learned concepts will be added to the ontology, newly learned words and phrases, to the lexicon; b) the enhanced ontology and lexicon will lead to the better quality TMRs produced by the ontological-semantic text analyzer; and c) the better quality TMRs will yield better results of the learning process, an early configuration of which has been reported in this paper.

We will continue to seek ways of including knowledge-lean (and, therefore, less labor-intensive) methods in the overall learning environment. However, the quality of the results of Experiments A and B was kept relatively low in a large part because we used knowledge-lean methods. In fact, the entire field of NLP has been favoring knowledge-lean methods for over a decade. This underscores the preference for coverage over depth and quality of description of individual language phenomena. Still, in a number of applications (e.g., machine translation) and tasks (e.g., part of speech tagging) sophisticated clustering methods used with large corpora yield acceptable results. The task we are pursuing does not seem to us to lend itself to solutions based on comparison.

Indeed, our goal is not to determine that the meaning of lexical unit A is closer to that of B than to that of C. It is to specify that meaning using an ontological metalanguage of properties and thus facilitate not only word sense disambiguation but also, using further ontological knowledge, semantic dependency determination, high-quality reference resolution and in general solutions to all meaning-dependent problems in NLP. At the

same time, we will experiment with other methods of statistical data processing after the data is mined from the web, with the immediate goal of reducing the quality gap between concepts and lexicon entries generated by human acquirers and automatically learned ones. In parallel, however, we will be looking for realistic knowledge-rich solutions to specific problems (e.g., we plan to incorporate our existing module processing diathesis transformations in English into Experiment B; had this been done already, only one sense of deport would be suggested by the system).

In parallel to work on unsupervised learning, we also plan to enhance our existing knowledge acquisition environment DEKADE [22][13], to include the option of presenting the results of automatic learning to human acquirers. This way we expect our work to contribute to the efficiency of human knowledge acquisition at an early stage.

**Table 8: Results of Combined Experiments A and C on Eleven Unknown Concepts, with clustering**

| Word | A | B | C | D |
|---|---|---|---|---|
| Brontosaurus | DINOSAUR | 0.607 | 0.607 | 0.715 |
| Cherimoya | FRUIT-FOODSTUFF | 0.607 | 0.646 | 0.637 |
| Depose | DEPOSE | 0.999 (ALL) | 0.999 (ALL) | 0.999 (ALL) |
| Diplodocus | DINOSAUR | 0.612 | 0.612 | 0.546 |
| Obey | OBEY | 0.518 | 0.646 | 0.518 |
| Pledge | PROMISE | 0.516 | 0.516 | 0.760 |
| Spartan | MILITARY-ROLE | 0.574 | 0.574 | 0.754 |
| Stegosaurus | DINOSAUR | 0.720 | 0.682 | 0.759 |
| Syrup | PLANT-DERIVED-FOODSTUFF | 0.646 | 0.573 | 0.760 |
| Triceratops | DINOSAUR | 0.643 | 0.721 | 0.849 |
| Wigger | SOCIAL-ROLE | 0.526 | 0.635 | 0.849 |

A: The targeted "correct" concept (existing in ontology). Repeated for clarity.

B: The best distance to the target, using attribute and relations (by OntoSearch comparison standards).

C: The best distance to the target, using relations and the best attribute value cluster (by OntoSearch comparison standards).

D: The best distance to the target, using relations only (by OntoSearch comparison standards). Repeated for clarity.

## References


[1] Beale, S., S. Nirenburg, K. Mahesh. Semantic Analysis in the Mikrokosmos Machine Translation Project. In *Proceedings of the 2nd Symposium on Natural Language Processing*, pp. 297-307, 1995.

[2] Beale, S., B. Lavoie, M. McShane, S. Nirenburg, T. Korelsky. Question Answering Using Ontological Semantics. In *Proceedings of ACL-2004 Workshop on Text Meaning and Interpretation*. Barcelona, Spain. 2004.

[3] Bernstein et al. How Similar Is It? Towards Personalizing Similarity Measures in Ontologies. In *Proceedings of the 13th European Conference on Information Systems.* 2005.

[4] Bikel, D. Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Proceedings of HLT*, 2002.

[5] Buitelaar, P., S. Handschuh and B. Magnini (eds.). In *Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population* (OLP). Valencia, Spain, August. 2004.

[6] Buitelaar, P. P. Cimiano, M. Grobelnik, M. Sintek. Ontology Learning from Text. Tutorial at ECML/PKDD, Porto, Portugal, October. 2005.

[7] Charniak, E., M. Berland. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the ACL*, pp. 57-64, 1999.

[8] Cimiano, P., G. Ladwig and S. Staab. Gimme' the context: Context-driven automatic semantic annotation with c-pankow. Proc. 14th WWW. ACM, 2005.

[9] Cimiano, P., J. Wenderoth, Automatically Learning Qualia Structures from the Web. In Proceedings of the ACL Workshop on Deep Lexical Acquisition, pp. 28-37, 2005.

[10] Clark, S., D.J. Weir. Class-Based Probability Estimation Using a Semantic Hierarchy. Computational Linguistics, 28(2), pp. 187-206, 2002.

[11] Curran, J., M. Moens. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 59-66, Philadelphia, PA, USA. 2002.

[12] Dempster, A., N. Laird, R. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion), *Journal of the Royal Statistical Society B, vol 39*. 1-38. 1977.

[13] English, J. DEKADE II: An Environment for Development and Demonstration in Natural Language Processing. Unpublished Master's Thesis, University of Maryland, Baltimore County. 2006.

[14] English, J., S. Nirenburg. Ontology Learning from Text Using Automatic Ontological-Semantic Text Annotation and the Web as a Corpus. In: Proceedings of Machine Reading AAAI Spring Symposium, 2007.

[15] Fikes, R., J. Jenkins, G. Frank. *JTP: A System Architecture and Component Library for Hybrid Reasoning*. Technical Report KSL-03-01, Knowledge Systems Laboratory, Stanford University, Stanford, CA, USA, 2003.

[16] Girju, R., D. Moldovan, Text Mining for Causal Relations, In *Proceedings of the FLAIRS Conference*, pp. 360-364, 2002.

[17] Java, A. et al. SemNews: A Semantic News Framework. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*. 2006.

[18] Lin, D. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th International Conference on Computational Linguistics, Vol 2*. 768-774. 1998.

[19] LingPipe, Alias I Inc, http://www.alias-i.com/. 2002.

[20] Manning, C. Language Learning: Beyond Thunderdome. In *Proceedings of Conference on Computational Natural Language Learning*, 2004.

[21] McShane, M., S. Nirenburg, S. Beale. An NLP Lexicon as a Largely Language Independent Resource. Machine Translation 19(2): 139-173. 2005.

[22] McShane, M., S. Nirenburg, S. Beale, T. O'Hara. Semantically Rich Human-aided Machine Annotation. Proceedings the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, ACL-05, Ann Arbor , pp. 68-75. June 2005.

[23] Navigli, R., P. Velardi. *Enriching a Formal Ontology with a Thesaurus: an Application in the Cultural Heritage Domain.* In *Proceedings of OLP-06*. 2006.

[24] Nichols, E., F. Bond, T. Tanaka, F. Sanae and D. Flickinger. *Multilingual Ontology Acquisition from Multiple MRDs.* In *Proceedings of OLP-06*. 2006.

[25] Nirenburg, S., V. Raskin. Ontological Semantics. SERIES: *Language, Speech, and Communication*, MIT Press, 2004.

[26] Ogata, N., N. Collier. Ontology Express: Non-Monotonic Learning of Domain Ontologies from Text. In *Proceedings of OLP*. 2004.

[27] Onyshkevych, B. Ontosearch: Using an ontology as a search space for knowledge based text processing. Unpublished PhD Dissertation. Carnegie Mellon University. 1997.

[28] Pantel, P., M. Pennacchiotti. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of Conference on Computational Linguistics / Association for Computation Linguistics (COLING/ACL-06)*. 113-120. 2006.

[29] Pedersen, T., and Bruce, R. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 197-207. 1997.

[30] Pustejovsky, J. *The Generative Lexicon.* Cambridge/London: MIT Press. 1995.

[31] Shutze, H. Automatic word sense discrimination. *Computational Linguistics 24, 1*, 97-124. 1998.

[32] Witten, I., F. Eibe. WEKA Machine Learning Algorithms in Java. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Chapter 8. Morgan Kaufmann Publishers. 2000.

[33] Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, 189-196. 1995.