Learning by Reading by Learning To Read

Sergei Nirenburg, Tim Oates and Jesse English
University of Maryland Baltimore County
{sergei,oates,english1}@umbc.edu

Abstract

Knowledge-based natural language processing systems learn by reading, i.e., they process texts to extract knowledge. The performance of these systems crucially depends on knowledge about the domain of language itself, such as lexicons and ontologies to ground the semantics of the texts. In this paper we describe the architecture of the GIBRALTAR system, which is based on the OntoSem semantic analyzer, which learns by reading by learning to read. That is, while processing texts GIBRALTAR extracts both knowledge about the topics of the texts and knowledge about language (e.g., new ontological concepts and semantic mappings from previously unknown words to ontological concepts) that enables improved text processing. We present the results of initial experiments with GIBRALTAR and directions for future research.

1. Introduction

This paper addresses a problem crucial for the success of semantic computing – automating acquisition of knowledge for knowledge-based systems. We are working on extracting knowledge from natural language texts. Our approach is to use our current capabilities for semantic analysis of texts to extract domain knowledge from the texts (learning by reading) and to feed a machine learning system to expand the knowledge resources (e.g., lexicon and ontologies) used to semantically analyze the texts (learning to read). The result is an ongoing spiral in which processing texts yields both domain knowledge and improved text processing.

At a somewhat iner level of detail, our GIBRALTAR system starts with the capabilities of semantic text analysis developed in the OntoSem system, an implementation of the theory of ontological semantics [1]. These capabilities are used by a learning system to generate, off of the meaning representations of large samples of text mined on the Web, knowledge structures describing the meanings of words and phrases that are not yet included in the knowledge resources of the semantic analyzer. At the next step, the newly learned knowledge structures (with or without a human validation step) are added to the semantic analyzer's knowledge resources, and the entire process is repeated. As a result of continuous operation of the system, the size of the knowledge base will grow, which will enhance the quality of the results of semantic analysis, which, in turn, will enhance the quality of the results of the learning step, and so on. In short, we are dealing with ongoing mutual bootstrapping of learning and text understanding.

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 [2]. 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 – [3] for meronymy, [4] for the qualia of the generative lexicon approach [5], and causal relations [6], 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 reported in [3] that uses essentially statistical methods for estimating selectional restrictions. Sources of knowledge acquisition include machine-readable dictionaries (e.g., [7]), thesauri (e.g., [8]), as well as text (e.g., [9], [10], [11]).

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.

In this paper, we present a) a brief description of the OntoSem semantic analyzer; b) the architecture of the GIBRALTAR system; c) our initial learning-by-reading experimentation and its results; and d) directions of future research and experimentation.

2. OntoSem

OntoSem 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., [12]), machine translation (e.g., [13]) 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 the JTP object-oriented reasoning system [14]. 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 experiment reported in this paper 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 for the results reported in this paper 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 [15]).

OntoSem knowledge resources have been developed 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. A high-level view of the architecture of the OntoSem analyzer is given in Figure 1.

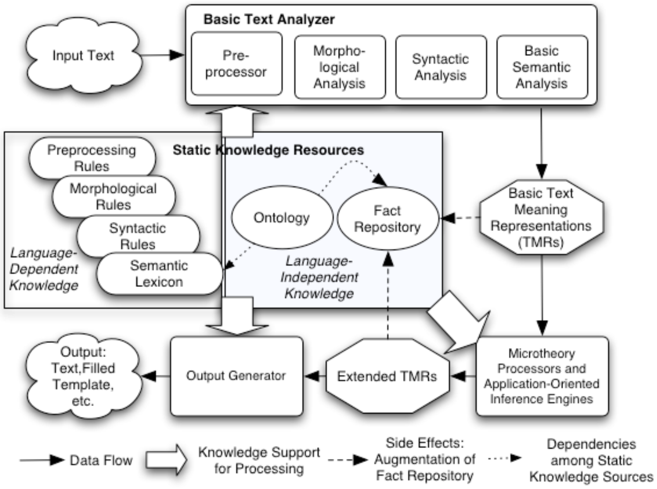
Basic Text Analyzer
Input Text
Pre-processor
Morpho-logical Analysis
Syntactic Analysis
Basic Semantic Analysis
Static Knowledge Resources
Preprocessing Rules
Morphological Rules
Syntactic Rules
Semantic Lexicon
Language-Dependent Knowledge
Ontology
Fact Repository
Language-Independent Knowledge
Basic Text Meaning Representations (TMRs)
Output: Text,Filled Template, etc.
Output Generator
Extended TMRs
Microtheory Processors and Application-Oriented Inference Engines
Data Flow
Knowledge Support for Processing
Side Effects: Augmentation of Fact Repository
Dependencies among Static Knowledge Sources

Figure 1. The Architecture of OntoSem.

3. The Experiment

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 [1], 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 there are words with missing senses.

For example, if the word "elephant" appears in a text and is not in the lexicon, we must a) determine that elephants have properties SIZE, COLOR, and WEIGHT, but not other properties in the ontology such as OCCUPATION or SALARY; b) determine that adult elephants weight between 3,500 and 7000 kg (and find similar expectations for values that fill the COLOR and SIZE slots); c) determine if a concept with similar properties and values exists in the ontology, creating a new one otherwise; and, finally, create a lexicon entry for "elephant" and link it to the ELEPHANT concept.

In view of the above, the experimental method that we use is as follows. We start with a list of words whose meanings will be learned by the system. This is done by using the OntoSem ontology 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.

For the initial experiment we selected a mixture of words for which there was no corresponding ontological concept in the current ontology and words for which an appropriate concept in the ontology existed but were removed for the duration of the experiment. Thus, for evaluation purposes, we had a "gold standard" in the latter case.

Next, we automatically acquire from the Web a corpus of sentences containing the target word and use OntoSem to generate TMRs for them. OntoSem is engineered to degrade gracefully in the face of unexpected input, so it is capable of semantically analyzing sentences with a number of unknown words by a) assuming that the unknown word's meaning corresponds directly to a non-existent ontological concept and b) unidirectionally applying relevant constraints listed in the ontological interpretations of the meanings of those words in input that are connected with the target word syntactically, thus hypothesizing the constraints on the meaning of the latter.

As a result of this stage, the system produces a set of pairs of property instances and their values. 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. Next, the result set is filtered to eliminate 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). Filters based on other heuristics are possible. The above process concentrates on properties with arity greater than 1. Unary properties, called attributes in OntoSem (e.g., MASS, COLOR, ADDRESS, as opposed to relations, such as AGENT or CAUSED-BY whose fillers are concepts, not literals), are currently mined from the web using the static knowledge resources of OntoSem (for query expansion) but not the analyzer itself. In other words, this component of the experiment is "knowledge-lean." Attribute values learned are appended to the property-value set for the target word. Our initial work on establishing the number of word senses for the target word as well as the methodology of combining the results of all experimentation are described in [17].

Once the cardinality of the property-value set for the target word reaches an empirically set threshold, this set is declared to be a candidate ontological concept. To find the appropriate place of the candidate concept in the ontological hierarchy, we compare it to concepts already existing in the OntoSem ontology. In our initial experiment [18], we used the OntoSearch algorithm [19] for this purpose. In subsequent experimentation we used a concept similarity metric of our own [20].

4. 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 [53] discusses two distinct methods of calculating similarity of concepts in an ontology: edge-based, and node-based (we will be using a combination of the two). Edge-based comparison has been implemented, for example, in the OntoSearch algorithm [19]. 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.

Table 1: Results of Experiment 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 target "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 similarity between the candidate and target concept.
E: The similarity between the candidate and the concept(s) the system determines as most similar to it.
F: The rank of the target (out of approximately 17,000 ontological concepts) in the automatically generated results
G: The distance between the target and the concept ranked as most similar to the candidate by the system, using relations only (using OntoSearch 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 this experiment are presented in Table 1. For each of 12 words, the table shows the target concept to which the word should be mapped, the number of clauses extracted than contain the word, the number of property/value pairs obtained by analyzing the clauses, and three different similarity values obtained from OntoSearch. The first of these is the similarity between the newly formed concept and the target; the second is the similarity between the newly formed concept and the existing ontological concept that is most similar to it as determined by OntoSearch; the third is the similarity between the target concept and the existing concept deemed most similar to the newly formed concept. In almost all cases, this third similarity value is quite high, and the first and second are quite similar, meaning that the system is doing a good job of finding the right place in the ontology for the new concept.

A different form of evaluation, one that we will pursue in the future, is to measure the coverage and accuracy of the semantic analyzer on texts before and after acquiring new words and the concepts that ground their semantics. Given these new words, the analyzer will have to fall back on heuristics less frequently and should be able to do a better job of both syntactic and semantic analysis. If the new words and concepts are useful, this will result in better output at the level of TMRs, which can be assessed by humans. If, on the other hand, the new words and concepts are harmful, the number and quality of TMRs will degrade over time.

5. **Our Place on the NLP and ML Maps**

Our approach to knowledge acquisition is based on two foundations – deep semantic analysis (interpretation) of the meaning of texts and automating the process of learning or acquiring this knowledge. What has been the attitude of the research community to meaning? The only commonly held opinion is that describing and manipulating meaning is very difficult. Different conclusions have been derived from this premise. Some people have decided to pursue NLP without meaning, some others have concentrated on non-NLP facets of intelligent systems (often assuming that NLP was already somehow available, e.g., [21] Lenat's (1995) Cyc project), and still others have studied disjoint meaning phenomena without attempting any comprehensive treatment (e.g., formal computational semanticists commonly study "glue" elements like quantifiers and conjunctions but do not pursue the rest of the lexicon).

In the spirit of avoiding meaning on the grounds that it is difficult (and therefore doing it will impede progress toward immediate results), NLP over the past 15 years or so has turned toward non-representational methods that rely on more or less sophisticated methods for comparing text strings with the goal of establishing the distance between them in some search space (e.g., latent semantic indexing, as in [22] Deerwester et al.1990). Another recent trend is to concentrate on non-NLP issues in the hope that they will benefit NLP in some way in the future: e.g., almost all work on the Semantic Web concerns representation languages, communication protocols, issues of trust, etc., completely abstracting away from the issue of the actual semantics of language; but without high-quality NLP, the Semantic Web will remain just a futuristic vision. A similar need for basic semantic analysis is evident in work on theorem-provers, which require a formal semantic representation to work on; however, the automated generation of that representation has not been pursued.

It is understandable that people have turned toward methods that can demonstrate short-term measurable improvement. A side effect of this shift of emphasis has been the interpretation of even low-quality results as useful – an influential paper in the field ([23] Church and Hovy 1993) was tellingly entitled "Good Applications for Crummy Machine Translation." Needless to say, the long-term goal of attaining computer understanding was, for all practical purposes, completely abandoned. Still, while this asemantic approach has acquitted itself well in cognitively simple fields (e.g., computer chess), ultimately, it will prove insufficient for high-end NLP.

Much of the lexical semantics research is either basically oriented at people (WordNet, e.g., [24] Fellbaum 1998), very narrow in its coverage (FrameNet, e.g., [25] Fillmore et al. 2001), very shallow in its semantic treatment (e.g., Beth Levin's [26] (1993) semantic classification of verbs based on syntactic behavior) or interested in generalizations rather than descriptions.

In the area of treating unexpected lexical input, a number of programs have attempted semantic analysis using scripts. Some of the first were FOUL-UP ([27] Granger 1977), which analyzed news stories, and NOMAD ([28] Granger 1983) which analyzed transmissions about ship and submarine activities. The latter produced better hypotheses regarding the meaning of unknown words because the narrow domain permitted highly specialized scripts to be written and leveraged. An enhancement to the strictly script-based approach was developed by Carbonell ([29]) (1981) in his POLITICS system, which analyzed texts about U.S. foreign policy using a combination of script-based and goal-based expectations. Most such research, however, grinded to a halt because it was deemed impossible scale up the complex knowledge structures. The TRUMP system, developed at GE [30] (Jacobs 1992) learns new words using productive derivational morphological processes and using a process of hypothesis and refinement to determine word meaning, the latter relying on finding multiple instances of the given word in a corpus and combining evidence about its co-occurrences. Like Jacobs, we plan to incorporate both morphological heuristics and hypothesis-refinement methods into our work. Other approaches to processing unknown words are either not implemented or apply to small domains (e.g., [31], [32] Kilbury et al. 1992; Barg and Walther 1998). A final aspect of research in unknown word processing is bypassing the need for onomasticons and gazetteers to understand proper nouns (e.g., [33] Bikel et al. 1999, [34] Mikheev et al. 1999, [35] Fleischman and Hovy 2002), which is a facility already incorporated into OntoSem. Research into automating the learning of subcategorization frames (e.g., [36] Carroll et al. 1998, [37] Brent 1991, [38] Manning 1993) will not directly inform our work since it typically uses few if any knowledge sources and is not applicable in a broad application domain. However, some related research that will be applicable to this project includes the findings that words occurring in certain types of syntactic configurations (conjunction structures, lists, appositives, and noun compounds) tend to belong to the same semantic class, and that statistical use of this information improves automated semantic class assignment [39-41] (Riloff and Shepherd 1997, 1999, Roark and Charniak 1998).

The automated learning of semantic roles incorporates available knowledge sources to varying extents. For example, McCarthy [42] (1997) develops stochastic methods for automatically positing semantic constraints, paying particular attention to the prerequisite of semantic disambiguation, a hurdle for many similar systems. Stevenson and Merlo [43] (2000) have implemented a system that automatically categorizes verbs into semantic classes, but it requires a tagged corpus, a resource on which we will not rely. Gildea and Jurafsky [44] (2002) have designed a system to automatically learn the semantic roles of unknown predicates but it makes two simplifying assumptions that our work will not: first, the system is provided with the correct syntactic frame for the verb and, second, the inventory of roles is far more limited than the ones provided in our ontology. While these and other such experiments have shown some promise, they largely circumvent the deep semantic analysis that we seek to pursue.

From an ML perspective, the most novel aspect of the proposed work is that the learner produces knowledge over time (entries in the lexicon and ontology) that significantly impacts what it learns in the future and its performance on the core task of NLU. Most ML systems produce outputs that in no way affect the operation of the underlying learning algorithm. Though lifelong learning has been a goal of the ML community for quite some time, few lifelong learning systems exist, and those that do tend do focus on transfer of learned knowledge across multiple, related tasks [45, 46]. After learning to perform one task well, the learner moves on to the second task, using what was learned in the first as, for example, a starting point. In contrast, our system will use learning to continuously improve its performance in a single task, namely, understanding text.

While computational learning by reading has been posited as a goal in the field (as in the Cyc project [21]), no real progress has been made because this process presupposes the ability to robustly extract meaning from texts, which no extant systems apart from OntoSem can do. Our vision of learning by reading involves dynamically and opportunistically identifying examples that are relevant to extending or refining our diverse knowledge base. Compare this with most supervised ML systems, which are given a fixed set of training examples [47], and active learning systems, which ask for specific examples to focus learning [48].

ML techniques have long been applied to developing or tuning information extraction (IE) systems, and more recent research has focused on learning to extract information from web pages [49-51]. Learning in this context invariably requires a fixed ontology to be known a priori, utilizes supervised learning techniques that rely on hand-labeled examples, and the output of the learner does not influence its future behavior. Co-training, which is using one learner to label examples for another, has been applied in this domain [51]; however, it does not produce knowledge that influences the learning capabilities of the other learner, it simply provides additional training data.

Ontologies are central to most attempts to express the semantics of text in a machine-tractable way. Fueled by interest in the Semantic Web, significant effort has gone into building methods for learning ontologies from free text. For example, the OntoWeb Consortium produced a survey of approaches to learning ontologies that contained several dozen methods [52]. Most of them are based primarily on statistical techniques, such as Latent Semantic Analysis (LSA) [22], that attempt to identify concepts (terms) in the text that are semantically related to concepts in an existing ontology. Other methods perform unsupervised clustering of terms to produce an ontology from scratch. The resulting ontologies have minimal internal structure. None of these techniques actively identify slot fillers for inductive learning to refine the ontology.

6. Discussion and Future Work

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 [16], 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.

References

[1] Nirenburg, S. and V. Raskin. Ontological Semantics. MIT Press. Cambridge, MA. 2004.

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

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

[4] 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.

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

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

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

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

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

[22] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R., Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6). 1990.

[23] Church, K. and Hovy, E. 1993. Good applications for crummy machine translation. *Machine Translation*. Vol. 8.

[24] Fellbaum, C. (Ed.). 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

[25] Fillmore, C. J., Wooters, C., & Baker, C. F. 2001. Building a large lexical databank which provides deep semantics. *Proceedings of the Pacific Asian Conference on Language, Information and Computation*. Hong Kong.

[26] Levin, Beth 1993. **Towards a Lexical Organization of English Verbs**. Chicago: University of Chicago Press.

[27] Granger, R. 1977. FOUL-UP: a program that figures out meanings of words from context. In *Proceedings of IJCAI.*

[28] Granger. R. 1983. The NOMAD system: Expectation-based detection and correction of syntactically and semantically ill-formed text. *Journal of Computational Linguistics* 9(3-4), 1983.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] Nirenburg, S. , D. Dimitroff, J. English and C. Pfeifer. Three Experiments on Mining the Web for Ontology and Lexicon Learning. Submitted to KDD-07.

[18] 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.

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

[20] English,J. and S. Nirenburg. Calculating Concept Similarity for Ontology Learning from Text. UMBC ILIT Technical Report, 2006.

[21] Lenat, D. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11). 199

[29] Carbonell, J. 1981. POLITICS: An experiment in subjective understanding and integrated reasoning. In R. Schank and C. Riesbeck, editors, *Inside Computer Understanding.* Lawrence Erlbaum,

[30] Jacobs, Paul S. Trump: A transportable language understanding program. International Journal of Intelligent Systems, 7:245-276, March 1992

[31] Kilbury, J., P. Naerger and I. Renz. 1992. New lexical entries for unknown words. *Theorie des Lexikons: Arbeiten des Sonderforschungsbereichs 282*, number 29.

[32] Barg, P. & Walther, M. 1998. Processing unknown words in HPSG. Proceedings of the Thirty-Sixth Annual Meeting of ACL and the Seventeenth International Conference on Computational Linguistics, Montreal.

[33] Bikel, D. M., Schwartz, R., & Weischedel, R. M. 1999. An algorithm that learns what's in a name. *Machine Learning* 34(1-3).

[34] Mikheev A., M. Moens and C. Grover. 1999. Named entity recognition without gazetteers. Proceedings of the l0th

Conference of the European Chapter of the Association for Computational Linguistics.

[35] Fleischman, M., & Hovy, E. 2002. Fine grained classification of named entities. *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*. Taipei, Taiwan, 267-273.

[36] Carroll, G., M. Rooth, D. Prescher and M. Light. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of EMNLP-3*, Granada, Spain.

[37] Brent, M. R. 1991. Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 209-214.

[38] Manning, C. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of 31st Meeting of the ACL*, 235-242.

[39] Riloff, E. and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-2)*.

[40] Riloff, E. and J. Shepherd. 1999. A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction. In *Journal of Natural Language Engineering* 5(2): 147-156.

[41] Roark, B. and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, 1110–1116.

[42] McCarthy, D. 1997. Word sense disambiguation for acquisition of selectional preferences. In Proceedings of the ACL/EACL 97 Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, Madrid, Spain, 52-61.

[43] Stevenson S. and P. Merlo. 2000. Automatic lexical acquisition based on statistical distributions. In *Proceedings of COLING 2000,* Saarbruecken, Germany.

[44] Gildea, D. and D. Jurafsky. 2002. Automated labeling of semantic roles. In *Computational Linguistics* 28(3): 245-288.

[45] [44] Thrun, S. A lifelong learning perspective for mobile robot control. In proceedings of the IEEE Conference on Intelligent Robots and Systems. 1994.

[46] Baxter, J. A Bayesian/information theoretical model of learning to learn via multi-task sampling. Machine Learning, 28(1). 1997.

[47] Mitchell, T. Machine Learning. McGraw Hill. 1997.

[48] Tong, S. and Koller, D. Support vector machine active learning with applications to text classification. In proceedings of Neural Information Processing. 2000.

[49] Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery. Learning to Construct Knowledge Bases from the World Wide Web. Artificial Intelligence, 118(1-2): 69-113. 2000.

[50] Krueger, B., Nillson, J, Oates, T., and Finin, T. Automatically generated DAML markup for semi-structured documents. In proceedings of the AAAI Spring Symposium on Agent-Mediated Knowledge Management. 2003.

[51] Muslea, I, Minton, S., and Knoblock, C. Hierarchical Wrapper Induction for Semistructured Information Sources. Autonomous Agents and Multi-Agent Systems 4(1/2): 93-114, 2001.

[52] Gómez-Pérez, A., and Manzano-Macho, A., A survey of ontology learning methods and techniques. 2003.

[53] Bernstein. How similar is it? Toward personalizing similarity measures in ontologies. In Proceedings of the 13th European Conference on Information Systems. 2005.