Calculating Concept Similarity Heuristics For Ontology Learning from Text

Jesse English and Sergei Nirenburg

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

Abstract

We present experimental results of an approach to learning ontological concepts from text. The ontological-semantic analyzer OntoSem and its knowledge resources – in particular, its NLP-oriented ontology and semantic lexicon – are used to dynamically create the feature values on which our learning approach is based. We expand upon our previously reported work, with emphasis given to development of a new metric for calculating similarity between two ontological concepts. The specific use for this metric in our approach is to compare an automatically generated candidate ontological concept with concepts already in the ontology, to find the best position for a new concept in the inheritance network of the ontology. Our long-term goal of bridging the knowledge acquisition bottleneck through "learning by reading" is assisted in this manner by facilitating the placement of acquired ontological concepts into an existing ontology.

1. Introduction An important long-term goal in facilitating AI is that of automated knowledge acquisition. A variety of methodologies have been employed to tackle this problem, for example, statistical methods in conjunction with part of speech tagging or semantic clustering of known and unknown words (Lin 1998), and generic pattern extraction for determining semantic relations (Pantel and Pennacchiotti 2006). We focus our knowledge acquisition approach on automatic extraction of text meaning using OntoSem (see Section 2) and using the resulting meaning representations as sources for empirically derived value sets for the set of properties (features) defined in the underlying OntoSem ontology. This approach can be seen as following two of the trends that Manning (2004) 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 differences 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 the 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 findings in automated knowledge acquisition with an emphasis on ontology learning. 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 (Buitelaar et al. 2005). The methods involved include different combinations of linguistic (knowledge-based) and statistical methods but mostly the latter. Work on extracting small subsets of such relations using largely statistical means has been reported (e.g., Charniak and Berland 1999 for meronymy, Cimiano and Wenderoth 2005 for the qualia of the generative lexicon approach (Pustejovsky 1995), causation (Girju 2002), among others). OntoSem, however, addresses the task of extracting knowledge about a large set of such relations using encoded knowledge as heuristics (cf. work by, e.g., Clark and Weir 2002 that uses essentially statistical methods for estimating selectional restrictions). Among the sources of knowledge acquisition are machine-readable dictionaries (e.g., Nichols et al.

2006), thesauri (e.g., Navigli and Velardi 2006), as well as text (e.g., Ogata and Collier 2004, Buitelaar et al. 2004, Cimiano et al. 2005). Our experiment uses open text but can be extended to treating MRDs and thesauri as special types of texts.

Most extant ontology learning methods operate at the level of textual strings, using sophisticated statistical analysis and clustering algorithms (optionally augmented by relatively shallow linguistic analysis) and thus requiring annotated training corpora. Our approach, by contrast, relies on a dynamically generated corpus of knowledge structures (TMRs) written in an ontological metalanguage, obtained through the operation of OntoSem, which relies on deep linguistic analysis strengthened by statistical algorithms operating over the ontology and the nascent TMRs. At present, the quality of automatically generated TMRs is not optimal. The plan 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 presents the ontological-semantic environment that underlies our NLP suite's operations; Section 3 briefly describes our experimental setting and presents the results from the previous work that we sought to improve; Section 4 introduces our latest method of clustering ontological concepts; Evaluation and comparison of results are presented in Section 5; And, finally, Section 6 is devoted to conclusions and future work.

2. OntoSem OntoSem (the implementation of the theory of Ontological Semantics; Nirenburg and Raskin 2004) 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., Beale et al. 2004), machine translation (e.g., Beale et al. 1995) 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 (Fikes et al., 2003). Text analysis relies on extensive 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 (we have recently developed a converter between this custom language and OWL, see Java et al. 2005).

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. A high-level view of OntoSem text processing is shown in Figure 1.

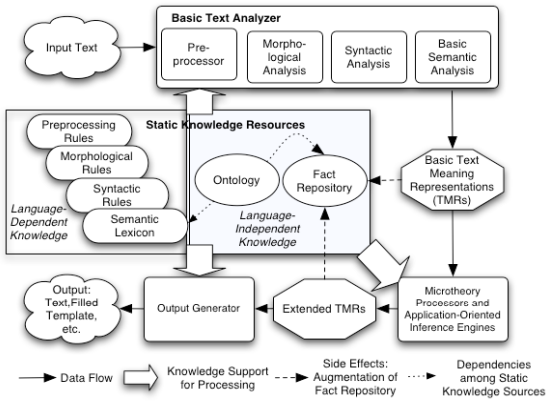

Figure 1. A High-Level View of the Architecture of OntoSem.

3. The Method. We made a simplifying assumption that the meaning of a word unknown to the system will be expressed as a univocal mapping to an ontological concept. Thus, the results of our experimentation were candidate ontological concepts, for which we needed to decide whether they should be added to the ontology (and if so, where in the ontological hierarchy this should occur). We start with a list of words whose meanings will be learned by the system. For the initial experiment we selected two words for which there was no corresponding ontological concept – *hobbit* and *pundit* – and two words for which an appropriate concept in the ontology existed – *CEO* (ontologically interpreted as PRESIDENT-CORPORATION) and *song* (interpreted using the ontological concept SONG). 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 this word and use OntoSem to generate their TMRs. OntoSem 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 those words in input that are connected with the unknown word to hypothesize 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. 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. In our initial experiment (English and Nirenburg 2007), we used the OntoSearch algorithm (Onyshkevych 1997) for this purpose. Table 1 shows an abridged version of the results as previously reported.

Word	Best Match	Selected Match	Difference
pundit	0.800	INTELLECTUAL 0.679	0.121 (15.1%)
ceo	0.900	PRESIDENT-CORPORATION 0.638	0.262 (29.1%)
hobbit	0.900	HUMAN 0.806	0.094 (10.4%)
song	0.800	SONG 0.800	0.000 (0%)

Table 1: Comparison metric results using OntoSearch.

4. Distance Calculation. As a means of improving the above results, we have developed a new similarity metric designed specifically to judge the distance between two concepts without calculating a minimal-weight path between the concepts (which is the method used by OntoSearch and which was an obvious drawback because such a path was assumed to exist in the ontology). In our current work the candidate concept is expected not to exist in the same ontology as the concepts it is being compared to.

As mentioned above, an ontological concept is a set of property/value pairs. Properties can be relations or attributes. Relations, such as THEME-OF, or CAUSES, refer (with restrictions) to other concepts in the ontology; attributes, such as COLOR and SIZE are one-place predicates that take their values from an ontologically defined value set (the content of the RANGE property of the ontological concept

describing the attribute). Properties of ontological concepts are not restricted to a single value; a concept may have a set of values for a given property.

The new metric has been designed to look at each property in the current OntoSem ontology and compare concepts on the basis of the values of each property. A vector, whose length is the size of the property inventory in the current ontology, is constructed, each element's value weighed between 0 and 1 (see explanation of weight assignment below).

In constructing a metric for comparing two concepts, we propose to take into account the number of property names that they have in common as well a measure of the similarity of value sets for the shared properties. We then propose to take a simple average these two values to produce a similarity value.. Let C_1 denote an existing concept and C_2, the candidate concept;, let P_t be the total number of properties defined by both concepts (*union* of properties) and P_s , the total number of properties shared by both concepts (*intersection* of properties); and let V_i denote the vector of computed value pairs for all values in C_1 and C_2 with property *I;* V_{iv} stand for combined results of value set comparisons for property *I; and* V_{gt} , for the total number of V_{iv} values greater than 0.0. We can then compute a value of the intersection of the sets of properties defined for each of the compared concepts as $P = P_s / P_t$ and the quality of the intersection of value sets for the defined properties as

$$V = \frac{\sum_{i=0}^{P_s} V_{iv}}{V_{gt}}$$

The simple averaging of the two values yields: Similarity = (P + V)/2.

We now turn to the issue of individual value comparisons. Here, different metrics must be developed for different types of property fillers (numbers, numerical ranges, symbols, ontological concepts and their sets, etc.) Table 2 shows a partial list of property comparisons, with a brief explanation of how a result is determined, Figure 2 specifies how numerical ranges are compared.

Value 1 Type	**Value 2 Type**	**Comparison Metric**
Text	Text	Case-insensitive character-by-character comparison
Text	Number (literal)	No match
Number (literal)	Number (literal)	Is Num1 within (Num2 x *TOLERANCE*) distance of NUM2?
Number (literal)	Number (relative)	Match
Number (literal)	Number Range	Is Num1 inside Range2? (Range2 is expanded by *TOLERANCE*)
Number (literal)	Concept	No match
Number (relative)	Number (relative)	Is Num1 within (Num2 x *TOLERANCE*) distance of NUM2?
Number (relative)	Number Range	Match
Number (relative)	Concept	No match
Numerical Range	Numerical Range	Are ranges equivalent? See Figure 2
Numerical Range	Concept	No match
Set	Concept	No match
Concept	Concept	Calculate distance to nearest common ancestor

Table 2: Value type comparison overview. *TOLERANCE* is defaulted to 10%.

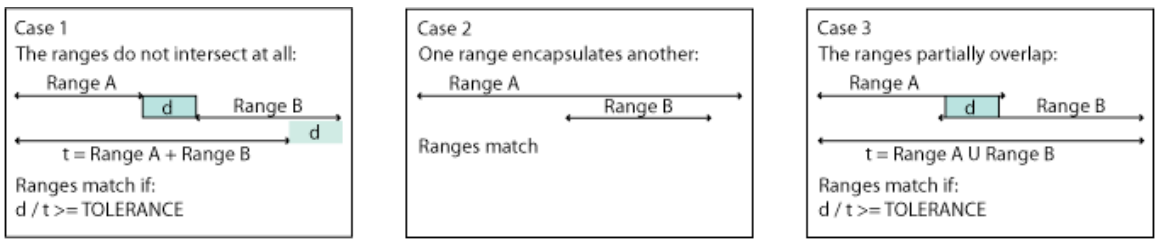

Figure 2: Three cases for numerical range comparison.

5. Evaluation. Using the new metric, we obtained the results summarized in Table 3. The last column in the table, **Improvement**, compares the results of using OntoSearch metric and our latest metric by comparing the distances between the system-generated and the human-determined best match using the two metrics (that is, the values in the **Difference** column of Tables 1 and 3).[1]

Word	Sample (no. of sentences)	Property Instances Extracted	Best Match	Desired Match	Differ-ence	Impro-vement
pundit	453	36	0.458	INTELLECTUAL (0.450)	1.75%	13.35%
CEO	552	23	0.448	PRESIDENT-CORPORATION (0.417)	6.92%	22.81%
hobbit	1458	157	0.520	HUMAN (0.493)	5.19%	5.21%
song	339	12	0.446	SONG (0.36)	17.71%	-17.71%

Table 3: New comparison metric results. The **Sample** column lists the number of sentences containing the word that were used as the corpus. Those sentences were processed and yielded the number of property-value set instances listed. These property-value set instances were combined into a candidate concept, and a best match was found for it using the new metric. The match selected by a human is listed in the **Desired Match** column. The **Difference** column shows the difference in the similarity scores between the system and the human user. The **Improvement** column compares new results with analysis using the OntoSearch metric.

In three out of four cases there was improvement. In the case of *song*, the new metric yielded worse results. This we attribute to the small size of the set of the automatically generated property/value instances that formed the candidate SONG concept. OntoSearch imparts more weight to subsumption properties (IS-A, SUBCLASSES, which are semantically weaker than other properties) than our current metric, and these properties were used predominantly to navigate the ontological hierarchy in the absence of more specific properties, resulting in a deceptively high score. The new metric, on the other hand, does not take these into account (very few actual sentences directly invoke subsumption relations), and therefore found very little useful information in SONG, resulting in a "decreased" quality of match (which is actually much more accurate than results using OntoSearch). Experiments reported in (English and Nirenburg 2007) suggested that the quality of the results improved with the increased number of automatically extracted and filtered properties increased. In the future, we will consider a threshold on the size of the set of the extracted property-value set pairs before even attempting to find the closest concept in the ontology to the candidate concept.

6. Discussion and Future Work. Experimentation reported here is a part of our ongoing work on learning by reading through life-long continuous mutual bootstrapping of the ontology and the ontology learning process. In such an approach, continuous improvement of OntoSem's static knowledge resources will gradually lead to larger and better property-value sets for candidate concepts due in part to larger text samples. We plan to run the experiment on a much larger set of unknown words and a much higher number of extracted property-value set pairs.

Simultaneously with the work reported here, we also use the OntoSem substrate to pursue automatic determination of the number of different senses for an unknown word as well as automatic empirical validation of the property-value sets already encoded in the ontology. Both these processes will assist the task of learning ontological concepts – by triggering a procedure to divide the candidate set of property-value set pairs into the number of candidate concepts corresponding to the suggested

[1] The absolute values for matches using the two metrics are not significant for this experiment. These values differ due to different penalty and normalization assumptions in the two approaches.

number of word senses for the original word and by improving the quality of the value sets to be matched, respectively.

We plan to develop means of determining whether an existing concept that was found to be sufficiently close to a candidate concept should be considered sufficient to describe the meaning of the original unknown word; or, alternatively whether the candidate concept should be added as a child or a sibling of the closest existing concept. We also plan to incorporate the learning functionality we are developing in the human knowledge acquisition toolset already in use in the OntoSem environment. Then an additional way to evaluate the utility of the automatic acquisition component will be in terms of timesavings for the knowledge acquirers (this approach is similar to machine-aided translation with post-editing).

References

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.

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

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

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

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.

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

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.

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.

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

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

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

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

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

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

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

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

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

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

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

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), pp. 113-120, 2006.

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