Toward Tomorrow's Semantic Web—An Approach Based on Information Extraction Ontologies

David W. Embley
Brigham Young University, Provo, Utah 84602, U.S.A.

Abstract: This position paper proffers the use of information-extraction ontologies as an approach to semantic understanding for the semantic web. From this perspective, it also issues challenges to the machine learning community to offer solutions for specific problems to aid in semantic understanding.

1 Introduction

Semantics is a grand challenge for the current generation of computer technology. It is the key for unlocking the door to personal agents that can roam the semantic web [BLHL01] and carry out sophisticated tasks for their masters.

The *American Heritage Dictionary* [AmH03] defines *semantics* as "the meaning or the interpretation of a word, sentence, or other language form." The keyword here is "meaning," but meaning requires understanding, and as Berners-Lee *et al.* state in their well known semantic web paper, "The computer doesn't truly 'understand' [anything]." They go on to say, however, that computers can manipulate terms "in ways that are useful and meaningful to the human user." This is a key point for semantic research in computing—we only have to manipulate symbols in ways that are meaningful and useful for human users. The illusion of understanding is sufficient if the symbol manipulation is good enough to obtain meaningful and useful results and good enough to allow us to trust the results at the level required for the application.

This position paper takes a tiny peck at the grand challenge of semantics by motivating a particular approach to dealing with semantics (Section 2); by giving some practical, real-world applications of the approach in the context of the semantic web (Section 3); and finally by pointing out challenges that need resolution, with a focus on problems whose resolution may involve machine learning (Section 4).

2 Motivation

Since computers do not truly "understand" what symbols mean, computer science researchers have the responsibility and the opportunity to creatively endow computers with the ability to perform useful tasks—indeed, to perform increasingly sophisticated useful tasks. How can we succeed in raising the level of sophistication required for tomorrow's applications?

Directing our discussion particularly to semantics, we first give some foundational material by defining data, information, knowledge, and meaning. Based on this foundation, we then state our central theme, which leads us to information extraction ontologies—the basis for the particular approach to "semantic understanding" proffered here.

2.1 Foundations

As a foundation, we give a variation of the definitions for data, information, knowledge, and meaning provided originally by Meadow (1992).

- *Data*: isolated attribute-value pairs.
- *Information*: data in a conceptual framework.
- *Knowledge*: information with a degree of certainty or community agreement.
- *Meaning*: data, information, or knowledge that is relevant or actuates.

These definitions help because they give us a working basis for "meaning," which we can take to be the results we want from "semantic understanding." Although meaning may well be "in the eye of the beholder," if we can, as Jim Gray said in a recent SIGMOD interview [Win03], "[take] data and [analyze] it and [simplify] it and [tell] people exactly the information they want, rather than all the information they could have," we will succeed in truly managing information.

Let's assume, as many do, that "meaning" for an individual is to be handled by personal software agents, which have access to knowledge both about their masters and about the world of interest to their masters, and concentrate on the foundational ideas needed to enable and actuate this assumption. Turning to "knowledge," which is next lower on the list, we observe that Meadow's definition of knowledge coincides with what most researchers call *ontologies*—agreed upon logical theories for an application domain, independent of any particular application [SMJ02]. Drilling further down in Meadow's definitions, we observe that logical theories are commonly conceptualized in a data model or conceptual framework, or, in Meadow's words, as information. Drilling to the bottom in Meadow's definitions, we arrive at isolated attribute-value pairs—data, the most fundamental concept for meaning.

In most of computer science, this foundation—the notion of data—is extremely weak. Our declarations of data typically only weakly classify data as *integer*, *real*, and *string* and provide only highly general operations over these classifications. To strengthen our foundation, we must have a much stronger notion of an attribute-value pair. We provide this stronger foundation through the use of data frames [Emb80]. A *data frame* "[encapsulates] the essential properties of everyday data items" such as currency, dates, weights, and measures. A data frame extends an abstract data type to include not only an internal data representation and applicable operations but also highly sophisticated representational and contextual information that allows a string that appears in a text document to be classified as belonging to the data frame. Thus, for example, a data frame for a birth date has regular expressions that recognize all forms of dates and regular expression recognizers for keywords such as "born," "born on," and "birth date" to distinguish a birth date from other dates such as the date of a meeting or a purchase date.

Our hypothesis is this: ontological conceptualization over data frames can increase shared understanding. The stronger foundation provided by data frames leads to richer, more understandable information, which, in turn, leads to a more solid bases for knowledge and the potential for increased understanding and more meaningful semantics.

2.2 Information Extraction Ontologies

We formalize ontological conceptualizations over data frames as *extraction ontologies*. In this conceptualization we fundamentally base an extraction ontology on its ability to recognize and classify value strings especially in semistructured and telegraphic text.

An *extraction ontology* is an augmented conceptual-model instance that serves as a wrapper for a narrow domain of interest such as car ads. The conceptual-model instance includes objects, relationships, constraints, and data-frame descriptions of strings for lexical objects. When we apply an extraction ontology to a document such as a web page, the ontology identifies objects and relationships and associates them with named object sets and relationship sets in the ontology's conceptual-model instance and thus wraps the page so that it is understandable in terms of the schema implicitly specified in the conceptual-model instance.

The ontological approach to writing wrappers directly addresses the hardest part of wrapper creation, which is to make a wrapper robust so that it works for all sites, including sites not in existence at the time the wrapper is written and sites that change their layout and content after the wrapper is written. Wrappers based on extraction ontologies are robust. Robust wrappers are critical: without them, we have to create by hand, or at best semiautomatically, a wrapper for every new web site encountered; with them, extracting information from new or changed web pages can be fully automatic. Ontology-based wrappers are an example of the kind of "intelligent" symbol manipulation that both gives the "illusion of understanding" and obtains meaningful and useful results.

3 Applications

Extraction ontologies can be used in many ways useful to semantic understanding and the semantic web.

Information Extraction. Our general approach to information extraction consists of the following steps. (See [ECJ+99] for full details.) (1) We develop an ontological model instance over the area of interest. (2) We parse this ontology to generate a conceptual schema and to generate rules for matching constants and keywords. (3) Given an applicable web page with multiple records (like classified ads), we

invoke a record extractor that separates an unstructured web document into individual record-size chunks [EJN99], gathers additional associated data linked on separate pages or factored into headers or footers [EX00], removes markup-language tags, and presents them as individual unstructured record documents for further processing. (4) We invoke recognizers that use the matching rules obtained from the data frames to identify potential constant data values and context keywords in the cleaned records. (5) Finally, we populate the generated conceptual schemas by using heuristics to determine which constants populate which concepts in the schema. These heuristics correlate extracted keywords with extracted constants and use cardinality constraints in the ontology to determine how to construct records. Once the data is extracted, we can issue queries using a standard query language (SQL or XQuery). To make our approach general, we fix the ontology parser, web record extractor, keyword and constant recognizer, and data record generator; we change only the ontology as we move from one application domain to another. We have applied extraction ontologies to many domains: apartment rentals, books, campgrounds, car-ads, cell phones, computer monitors, computer software, countries, course catalogs, digital cameras, games, gems, genealogy, jewelry, jobs, movies, music, obituaries, personals, pharmaceutical drugs, restaurants, stocks, and more.

Semantic Web Page Annotation. Semantic web page annotation is an immediate consequence of ontology-based information extraction. We extract directly into an ontology, and we can retain links to original web pages. From this intermediate form, we can generate annotations for semantic web pages in any form we wish. We have, for example, generated RDF specifications [Cha03], and we are planning to generate OWL specifications [Din05].

High-Precision Classification. Based on extraction ontologies, we have proposed a technique for high-precision recognition of web documents that apply to an ontologically specified domain [ENX01]. High-precision classifiers determine not only whether a document, such as a listing of classified ads in a newspaper, contains items of interest for a predefined application ontology, but also whether particular elements of interest are present in the document. It should be clear that if we can extract the basic information in a document relative to an application domain, then we can apply heuristic measures over these extracted values to determine whether the document is sufficiently similar to documents expected in the application domain and thus do high-precision classification.

Free-Form Semantic Web Queries. Given web pages with semantic annotation and a reasonably detailed free-form user query, it should be possible to extract information from the query, find the best matching semantic web ontology, embed the query in the ontology, determine the select-project-join requirements of the query with respect to the ontology, and provide a reasonable answer to the query [EK85]. For example, a user might request the following.

> Tell me about cruises on San Francisco Bay. I'd like to know scheduled times, cost, and the duration of cruises on Friday of next week.

An extraction ontology built for travel information should be able to recognize and extract "cruises", "San Francisco Bay", "scheduled times", "cost", "duration", "Friday", and "next week", all with respect to the ontology. The system should then be able to determine that "San Francisco Bay," "Friday," and "next week" are selection constants or can be turned into selection constants, find the join paths in the ontology that connect all the recognized concepts, and realize that the projection requirements are for "scheduled times", "cost", and "duration", which should be the results of the request. Since the query responses in such a paradigm are not likely to always provide exactly what the user wants, the system can and should provide both links to the original web sites that supplied the information for the semantic annotation and a ranked list of alternative answers with respect to different ontologies and web sites.

Task Ontologies for Free-Form Service Requests. Similar to answering free-form semantic web queries, it should be possible to also provide services for everyday tasks such as scheduling appointments, selling, buying, and so forth [AM05]. The approach to this challenge centers around a task ontology. A task ontology can be thought of as having two component ontologies: (1) a domain ontology that defines concepts in a domain of a task and (2) a process ontology that defines processes for doing tasks. Given a free-form, user-specified task request such as

> I want to see a dermatologist next week; any day would be OK for me, at 4:00. The dermatologist must be within 20 miles from my home and must accept my insurance.

the system should (1) use an extraction ontology to recognize keywords, keyword phrases, constants, and computable constants in the request, (2) compare the extracted information against available task ontologies to find the most applicable ontology, (3) discover and obtain missing required information

either from system repositories or from the user, (4) assemble software components to do the task, and (5) perform the task, negotiating as necessary with the user to complete the task.

Schema Mapping for Ontology Alignment. Automatic schema mapping for ontology alignment is a challenging task. The processes is especially challenging when the concepts in the two ontologies do not align one-to-one. We have studied the application of extraction ontologies to automate 1:n and n:1 as well as 1:1 schema-mapping techniques for populated ontologies. Extraction ontologies appear to improve mapping accuracy as well as provide a bases for non-1:1 mapping discovery [EXD04].

Record Linkage. The record-linkage problem arises when we try to merge populated ontologies. How do we know whether two objects are the same? Record-linkage algorithms typically use heuristics to determine object identity. One technique that can work for web pages is to use an extraction ontology to retrieve identifying information for an object and then use this extracted information to heuristically determine whether two objects are the same. In an experiment, we used this technique to determine whether two Google-returned citations for a person-name query refer to the same person [AKE04].

Ontology Discovery. Although ontologies and the semantic web are offered as a potential solution to today's information-explosion problems, creating ontological descriptions for real-world information is nontrivial. If we could automate the process, we could significantly improve our chances of making the semantic web a reality. While understanding natural language is difficult, tables and other structured information make it easier to interpret new items and relations. In [TELN03] we present an approach to generating ontologies based on table analysis. We thus call our approach TANGO (Table ANalysis for Generating Ontologies). Based on an extraction ontology, TANGO attempts to (1) understand a table's structure and conceptual content, (2) discover the constraints that hold among concepts extracted from the table, (3) match the recognized concepts with ones from a more general specification of related concepts, and (4) merge the resulting structure with other similar knowledge representations. TANGO is thus a formalized method of processing the format and content of tables that can serve to incrementally build a relevant reusable conceptual ontology.

4 Challenges

Although there are many challenges—indeed, there are grand challenges—we focus here only on a few specific challenges and particularly those that appear (to an outsider) to be in the realm of machine learning. There are likely many others.

Web Page Understanding. Given a web page and extracted data that is roughly only 85% accurate, generate a page grammar for the page. This would enhance extraction (1) by enabling better recall because additional, initially unrecognized values could be extracted, (2) by enabling better precision because many false positives could be discarded, and (3) by generating a fast extraction processor for the page for subsequent extraction either from sibling pages or from pages in which the data (but not the structure) has changed.

Universal Rules for Schema Matching. The work in [DMD+03] learns domain-specific rules for matching schemas. In [EJX02], an attempt was made to learn "universal" rules for concept name matching and instance-based matching. The challenge is to use machine learning to develop universal rules that are useful across all domains based on training data obtained for only some domains.

Boundaries of Usefulness. Find the boundaries of usefulness for machine learning. When should the technique *not* be used? When is it better to hard code heuristics or use probabilistic reasoning or some other technique?

Application to Significant Problems. Can we use machine learning to match free-form semantic web queries and service requests to domain and task ontologies? ... to find universal rules for record linkage? ... to discover constraints among concepts in semi-structured data? ... to separate records in multiple-record documents and consolidate records with factored and linked data? ... to enhance data-instance recognizers? ... to learn universal thresholds or application-characteristic-dependent thresholds? ...

We can, and indeed must, meet the challenges for semantic understanding for the semantic web. Otherwise, the semantic web will fail.

Acknowledgments: This material is based upon work supported by the National Science Foundation under grant No. IIS-0083127.

References

[AKE04] R. Al-Kamha and D.W. Embley. Grouping search-engine returned citations for person-name queries. In *Proceedings of the ACM Sixth International Workshop on Web Information and Data Management (WIDM 2004)*, pages 96–103, Washington, DC, November 2004.

[AM05] M. Al-Muhammed. Ontology aware software service agents: Meeting ordinary user needs on the semantic web. PhD Dissertation Proposal, 2005.

[AmH03] American heritage dictionary. education.yahoo.com/reference/dictionary/, 2003.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 36(25), May 2001.

[Cha03] T. Chartrand. Ontology-based extraction of RDF data from the world wide web. Master's thesis, Brigham Young University, Provo, Utah, March 2003.

[Din05] Y. Ding. Semantic annotation using data-extraction ontologies. PhD Dissertation Proposal, 2005.

[DMD+03] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *VLDB Journal*, 12:303–319, 2003.

[ECJ+99] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.

[EJN99] D.W. Embley, Y.S. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 467–478, Philadelphia, Pennsylvania, May/June 1999.

[EJX02] D.W. Embley, D. Jackman, and L. Xu. Attribute match discovery in information integration: Exploiting multiple facets of metadata. *Journal of the Brazilian Computing Society*, 8(2):32–43, November 2002.

[EK85] D.W. Embley and R.E. Kimbrell. A scheme-driven natural language query translator. In *Proceedings of the 1985 ACM Computer Science Conference*, pages 292–297, New Orleans, Louisiana, March 1985.

[Emb80] D.W. Embley. Programming with data frames for everyday data items. In *Proceedings of the 1980 National Computer Conference*, pages 301–305, Anaheim, California, May 1980.

[ENX01] D.W. Embley, Y.-K. Ng, and L. Xu. Recognizing ontology-applicable multiple-record web documents. In *Proceedings of the 20th International Conference on Conceptual Modeling (ER2001)*, pages 555–570, Yokohama, Japan, November 2001.

[EX00] D.W. Embley and L. Xu. Record location and reconfiguration in unstructured multiple-record web documents. In *Proceedings of the Third International Workshop on the Web and Databases (WebDB2000)*, pages 123–128, Dallas, Texas, May 2000.

[EXD04] D.W. Embley, L. Xu, and Y. Ding. Automatic direct and indirect schema mapping: Experiences and lessons learned. *SIGMOD Record*, 33(4):14–19, December 2004.

[Mea92] C.T. Meadow. *Text Information Retrieval Systems.* Academic Press, San Diego, California, 1992.

[SMJ02] P. Spyns, R. Meersman, and M. Jarrar. Data modeling versus ontology engineering. *SIGMOD Record*, 31(4):12–17, December 2002.

[TELN03] Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, and G. Nagy. Ontology generation from tables. In *Proceedings of the 4th International Conference on Web Information Systems Engineering*, Rome, Italy, December 2003. 242–249.

[Win03] M. Winslett. Jim Gray speaks out. *SIGMOD Record*, 32(1):53–61, March 2003.