Extending Metadata Definitions by Automatically Extracting and Organizing Glossary Definitions

Eduard Hovy[1], Andrew Philpot[1], Judith Klavans[2], Ulrich Germann[1], Peter Davis[2], Samuel Popper[2]

[1] Digital Government Research Center
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292

[2] Digital Government Research Center
Department of Computer Science
Columbia University
535 West 114th Street, MC 1103
New York, NY 10027

Abstract

Metadata descriptions of database contents are required to build and use systems that access and deliver data in response to user requests. When numerous heterogeneous databases are brought together in a single system, their various metadata formalizations must be homogenized and integrated in order to support the access planning and delivery system. This integration is a tedious process that requires human expertise and attention. In this paper we describe a method of speeding up the formalization and integration of new metadata. The method takes advantage of the fact that databases are often described in web pages containing natural language glossaries that define pertinent aspects of the data. Given a root URL, our method identifies likely glossaries, extracts and formalizes aspects of relevant concepts defined in them, and automatically integrates the new formalized metadata concepts into a large model of the domain and associated conceptualizations.

1. Background: The EDC System

The Digital Government Research Center (DGRC) has been building a data delivery system called the Energy Data Collection (EDC) project (Ambite et al. 02; Philpot et al. 02; Hovy et al. 01). The system delivers to users data about the prices and volumes of petroleum product sales, as collected and recorded by the Energy Information Administration (EIA; see http://www.eia.doe.gov), the Bureau of Labor Statistics (BLS), the Census Bureau, and the California Energy Commission (CEC). The system currently provides access to over 58,000 data tables provided by these agencies. These tables are stored in many formats, including Microsoft Access databases, flat files of numbers, html pages, PDF files, etc.

To provide dynamically planned access to so many non-homogeneous databases, the system employs the query planner SIMS (Arens et al. 96) that decomposes a user's query into smaller manageable plan of subqueries (Ambite and Knoblock 00), expresses them in SQL, retrieves the appropriate data, and recombines it as required. In order to determine how to decompose the query, and to determine where to locate each specific type of data, SIMS employs a formal model of the domain in which each type of data is represented by an entity (which we call a concept) appropriately taxonomized in relation to other similar concepts. The domain model contains approximately 500 concepts, arranged in some dozen small taxonomies, each subtaxonomy modeling one pertinent aspect of the data in the Energy Domain. For example, one taxonomy models locations (states, areas, districts, etc.), another models units of measure (gallons, barrels, etc.), and a third models frequency of measurement (monthly, biweekly, etc.).

The leaf concepts of each subtaxonomy directly and fully express the semantics of some aspects of the data in one or more databases in the collection. The formal equivalence holding between the leaf concepts and the data ensures that the SIMS planner can confidently plan with the concepts and return the data they indicate as responsive to the user's query.

In addition to helping SIMS plan its queries, the domain model has another function. Using a browser interface provided by the EDC system, the user can explore the taxonomies of the domain model in order to become familiar with the specifics of the data collections. Should two databases have slightly different

definitions for the *Mid-Atlantic region*, for example, there will be two corresponding concepts in the subtaxonomy modeling *Areas*, with associated textual descriptions and possibly hyperlinks to associated online text. Such details allow the user to ensure that he or she is thoroughly familiar with exactly the data being requested from the system.

The domain model is the minimal collection of concepts required to define the data. Such minimalism is generally appreciated by the system builder and domain experts, but can lead to problems. Since the domain model contains no higher-level generalization concepts, it has no overarching organizing structure to relate its subtaxonomies to each other. One consequence is that non-expert users, who generally do not know the domain terminology, can have difficulty finding what they need or orienting themselves in the domain model. Another consequence is that adding new concepts that represent hitherto unmodeled aspects of (new) databases is difficult, since there are no obvious attachment points in the subtaxonomies.

We have therefore embedded the domain model into a very large general-purpose taxonomy of some 110,000 concepts called the Omega ontology (Hovy et al. 03). Omega, a successor to the SENSUS ontology built at ISI (Knight and Luk 94), likewise contains Princeton's WordNet (Fellbaum 98), but includes also New Mexico State University's Mikrokosmos/OntoSem (Mahesh and Nirenburg 95) and a wholly new upper structure. Using a partially automated linking process described below, the domain model concepts have been indexed into Omega, allowing the casual user to provide almost any English word to the system's ontology browser in order to locate closely associated domain model concepts. Omega concepts are connected to the domain model with what we call Generally Associated With (GAW) links. In contrast to the links from the domain model to the databases, which are formal and exact, the GAW links are less exact, more in the nature of free association, in order to connect the user's possibly ambiguous or slightly misphrased term choices to the right places.

Embedding the domain model into Omega has a second major benefit. When new databases are added to the system, new metadata concepts must be added to the domain model as well. When a new concept is a close variant of an existing domain concept, this is not a problem, but when it is rather different, or perhaps something altogether outside the current domain, the modeler requires some appropriate position for it. Omega, being a very general domain-independent taxonomy, provides a first approximation.

We describe in (Hovy et al. 02) experiments in developing automated methods of linking domain concepts into Omega (or, at that time, SENSUS), using heuristics that compare concepts' names, textual definitions, and hierarchical organization across taxonomies. In (Philpot et al. 02) we describe AskCal, a natural language interface that converts the user's English question into a SIMS query, possibly via a cascaded menu interface based upon the domain model contents. Work is underway to create equivalent AskCal interfaces for Spanish and Chinese.

In this paper, we address a problem at the center of all efforts to provide access to multiple heterogeneous databases: how can one (help) automate the process of domain modeling? In particular, given a new database, how can one produce, even tentatively, new concepts for the domain and/or Omega taxonomies to express unmodeled aspects of the new data? If one can succeed in doing so, the task of the domain modeler is considerably facilitated; modeling becomes a process of checking the postulated concepts for definitional correctness, appropriate placement in the taxonomy, and correct cross-linkage.

2. Growing the Domain Model Automatically

In this section we describe our method, still under development, that involves components from its precursor GlossIT (Klavans et al. 02). The method is based on the assumption that new databases are accompanied by textual material in the form of online glossaries (Muresan et al. 03). Frequently, experts create glossaries as a means of specifying carefully and explicitly their work. Usually, glossary terms are not general-purpose English words but very specific variants of them, as used by the experts in their particular endeavor. These definitions thus tend to exhibit a rather standardized structure, something a text analysis program can exploit. In the rest of this section we describe each stage of our method.

2.1 Glossary Identification--GetGloss

The first step of our process involves identifying glossary files within a website, given a root URL. GlossIT contains the GetGloss module that performs this function. Recursively, GetGloss visits each page under a target URL, down to depth five. A filter inspects each page. Likely glossaries are accepted and their URLs are added to the recursive search list.

The principal technical problem is creating a high-precision filter. Variations in content, style, and html usage must be handled (for example, non-standard usage, such as using " " and " " to delineate headwords, occurs frequently). The approach taken was page categorization (into the categories glossary or not-glossary), using a set of hand-crafted rules that searched for content keywords, certain html constructs, and other identifying marks.

To test the adequacy of these rules, we also categorized using several algorithms provided in the Rainbow text categorization toolkit (McCallum 96), as well as the Ripper system (Cohen 95). As described in Section 3.1, Rainbow's Support Vector Machine module performed extremely well. However, care must be taken to train the modules on pages with the same characteristics as the target.

2.2 Concept Formation

The task of concept formation is to identify and delimit the text describing each concept, locate within each text string specific aspects of interest, extract and convert them into the format required, if appropriate cross-index terms, and produce a formalized concept definition frame. For example, the html

```
<b><a name=benzene">Benzene</a>
(C<sub><font size="1">6</font></sub>H<sub><font size="1">6</sub></font>):</b>
An aromatic hydrocarbon present in small  proportion in some crude oils and made
commercially  from petroleum by the catalytic reforming of naphthenes in petroleum
naphtha. […]
```

is converted into the frame

```
(DEFCONCEPT |benzene|
 :lex ("benzene")
 :formula |C6H6|
 :direct-superclass |hydrocarbon|
 :properties (|aromatic|)
 :definition "An aromatic hydrocarbon present in small  proportion in some crude oils and
made commercially  from petroleum by the catalytic reforming of naphthenes in petroleum
naphtha.")
```

which is then passed on for taxonomizing.

This task is accomplished by Information Extraction techniques. In the first step, we extract the plain definition text from the source and, at the same time, exploit typographic information, which is preserved in the html markup to perform some coarse chunk parsing using regular expressions. In the next step, we perform a full parse of the defining sentence, using Charniak's statistical parser *parseIt* (Charniak 99). The use of a full parser may actually not be necessary (cf. Alshawi 87; Barnbrook and Sinclair 95) but makes it easier for us to identify the *head* of the definition, which usually represents the supertype. Since *parseIt* was trained on full sentences rather than the idiosyncratic format of glossary entries, we rephrase the definition as a full sentence: *Benzene is an aromatic hydrocarbon…* As a rule of thumb, we assume that the *last* noun in the leftmost NP under the VP node corresponds to the closest supertype. Leading adjectives ("aromatic") are interpreted to be properties that distinguish a concept from (at least some of) its siblings. Trailing phrases (PPs, relative clauses, etc.) are currently not utilized.

Even though the majority of the glossary entries follow this pattern, many do not. Consider, for example, the following glossary entry:

> **Apparent consumption**: coal production plus imports of coal, coke, and briquets minus exports of coal, coke, and briquets plus or minus stock changes.

Obviously (to the human, but not to the system), apparent consumption is ***not*** some sort of production. It is the result of some calculation. In addition, it is not clear what a corresponding concept ideally should look like. Should it be tied to coal? Should the ontology contain a concept for the *apparent consumption* of every commodity? Or should generativity be part of a dynamic ontology design that has an abstract concept prototype that can be applied to all kinds of commodities (apparent consumption of *X* is production of *X* plus imports of *X* plus/minus the respective stock changes). On the other hand, once these questions are resolved, pattern matching can be used to recognize and interpret such formulas. Our work currently puts the emphasis on extracting subtype-supertype relations in order to facilitate faster development of large scale and domain specific taxonomies. Evaluation results are given in Section 3.2.

Prior work exists both within the project, in the ParseGloss component of GlossIt (Klavans and Whitman 01) and with other material (Klavans and Muresan 00). There is also a significant amount of work on extracting information from dictionaries (Barnbrook and Sinclair 95; Wilks et al. 96).

2.3 Concept Placement

The task in this stage is to locate for a given concept, represented by the frame, the most appropriate attachment point(s) in the domain model and/or Omega. Because this operation involves real understanding of the meanings of concepts, human verification is required. However, as shown in past research on cross-ontology alignment (Resnik 93; Agirre et al. 94; Knight and Luk 94; Hovy 98), automated methods can with some accuracy suggest likely parent or sibling concepts, given enough information to start with. In fact, attempting to learn domain models through concept extraction and clustering both automatically (Doan et al. 00; Hovy et al. 01) or semi-automatically (Knoblock et al. 00) is an exciting and ambitious area of research.

By and large, the alignment methods each include one or more heuristics that compare some aspect of an input concept against that of each concept in the target taxonomy and produces a closeness rating. A subsequent module then combines the ratings of various heuristics under some weighting, whose parameters are usually automatically learned by considering good and bad example alignments.

Three classes of alignment heuristic have emerged: textual (comparing concept names, concept definitions, etc.), data value and format (comparing value orthography, types, and restrictions, etc.), and taxonomic (comparing position with regard to known aligned concepts, comparing degrees of cluster dispersal, etc.). These heuristics are all rather parameterizable, unfortunately, and hence require considerable tuning or machine learning for optimal behavior. For example, a fairly extensive parameter sweep study to determine good settings for text matches of definitions (including stop words or not, demorphing words or not, taking into account word sequence or not, etc.) is reported in (Hovy et al. 01).

For the current study, we applied our library of text and taxonomy matching heuristics, associating with every candidate concept a set of ratings. We presented this set of features to a decision tree learner as in (Hovy et al. 03) tuning the model by training with a set of hand-aligned positive examples. We matched candidate concepts separately against the EDC domain model and Omega as well as combined.

2.4 Display and Verification

The newly acquired concepts, located in their proposed locations and connected to the appropriate value concepts, are displayed in the browser for the user.

3. Experimental Results

3.1 GetGloss Results

We tested GetGloss on 2608 files collected under www.census.gov, and compared it to several categorization algorithms (Naïve Bayes, Support Vector Machine, k-Nearest Neighbor, and Probabilistic

Indexing) from the Rainbow package. Table 1 provides Recall, Precision, and F1 scores for these algorithms, first on glossary files and then on non-glossary files.

Algorithm	Recall	Precision	F1	Recall	Precision	F1	Accuracy
GetGloss	1.000	0.517	0.682	0.994	1.000	0.997	0.994
Naïve Bayes	1.000	0.162	0.279	0.957	1.000	0.978	0.958
SVM	0.944	0.895	0.919	0.999	0.999	0.999	0.999
KNN	0.000	0.000	0.000	1.000	0.992	0.996	0.992
ProbIndex	0.222	0.400	0.286	0.997	0.994	0.995	0.991

Table 1. Results of GetGloss, using various algorithms on 2608 files under www.census.gov. Columns 2 to 4: scores on glossary files; columns 5 to 7: non-glossary files.

In this test GetGloss performed well, since this is the set of documents it was developed on. We also applied the algorithms to documents extracted from the roots www.epa.gov and www.bls.gov. In this case, the Support Vector Machine outperformed every other algorithm.

3.2 Concept Formation Results

Out of a glossary with 2182 entries, the system postulated superconcepts for 1447 terms. 284 of these terms are already defined in Omega (that is, there is at least one concept in Omega that is mapped to a lexical item that is identical in spelling to the term in the glossary). The postulated superconcepts of 102 of these terms matched one of the superconcepts for the corresponding concept in Omega. In other words, a rough, automatic evaluation of the current system indicates a precision of roughly 36%.

3.3 Concept Placement Results

Placing concepts into the Omega taxonomy follows the methodology of (Hovy et al. 03). Since our semi-supervised approach anticipates that a human will need to choose from various proposed candidate alignment points, the decision tree is deliberately parameterized to have high precision at the possible cost of very low recall. We constructed two decision trees using C4.5. The first attempted to classify a pair of concepts as equivalent, super/sub (the EIA concept is a child of some Omega concept), or negative. The second, recognizing that negative instances come from more than source, allowed classification of positives into equivalent and super cases, but also negatives into "false cognate" (similar names but conceptual mismatch), "random" negative, and negatives which essentially came from the decision tree failing to match anything. Unfortunately, the results are disappointing. The statistics deliver between 60% and 70% precision and recall for positives, and for negatives over 90% for both measures.

4. Conclusion

Domain modeling is a time-consuming and difficult process. Automatically 'scraping' concepts from under a root website, formalizing them, and proposing locations in a domain model can significantly minimize the system builder's and domain modeler's work. Even if the concept placement (the weakest step of the method at present) is not all that accurate, a drag-and-connect operation via the browser is much faster and more natural than laboriously locating parent concepts by hand via a text editor.

Our current work involves semi-automated domain model creation via database mining. Ultimately, we anticipate that the combination of database and text/glossary mining techniques will significantly improve the speed and effectiveness of domain model construction.

References

Agirre, E., X. Arregi, X. Artola, A. Diaz de Ilarazza, and K. Sarasola. 1994. Conceptual Distance and Automatic Spelling Correction. *Proc. of Workshop on Computational Linguistics for Speech and Handwriting Recognition*.

Alshawi, H. 1987. *Memory and Context for Language Interpretation*. Cambridge: Cambridge University Press

Ambite, J.L. and C.A. Knoblock. 2000. Flexible and Scalable Cost-Based Query Planning in Mediators: A Transformational Approach. *Artificial Intelligence Journal* 118 (1–2).

Ambite, J.L., Y. Arens, L. Gravano, V. Hatzivassiloglou, E.H. Hovy, J.L. Klavans, A. Philpot, U. Ramachandran, K. Ross, J. Sandhaus, D. Sarioz, A. Singla, and B. Whitman. 2002. Data Integration and Access: The DGRC's EDC Project. In W. Mcver and A.K. Elmagarmid (eds), *Advances in Digital Government*. Dordrecht: Kluwer.

Arens, Y., C.A. Knoblock and C.-N. Hsu. 1996. Query Processing in the SIMS Information Mediator. In A. Tate (ed), *Advanced Planning Technology*. Menlo Park: AAAI Press.

Barnbrook, G. and J. Sinclair. 1995. Parsing Cobuild Entries. In J. Sinclair, M. Hoelter and C. Peters (eds). *The Languages of Definition: The Formalisms of Dictionary Definitions for Natural Language Processing*. Studies in Machine Translation and Natural Language Processing Vol. 7. Luxembourg: Office for Publications of the EC.

Charniak, E. 1999. *ParseIt* statistical parser. Available from ftp://ftp.cs.brown.edu/pub/nlparser/.

Cohen, W. 1995. Fast Effective Rule Induction. *Proceedings of the International Conference on Machine Learning*.

Doan, A., P. Domingos, and A. Levy. 2000. Learning Source Descriptions for Data Integration. *Proceedings of a Workshop at the Conference of the American Association for Artificial Intelligence (AAAI)*.

Fellbaum, C. 1998. (ed.) *WordNet: An On-Line Lexical Database and Some of its Applications*. MIT Press.

Hovy, E.H. 1998. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. *Proceedings of LREC*. Granada, Spain.

Hovy, E.H., A. Philpot, J.L. Ambite, Y. Arens, J. Klavans, W. Bourne, and D. Saroz. 2001. Data Acquisition and Integration in the DGRC's Energy Data Collection Project. *Proceedings of dg.o 2001*. Los Angeles.

Hovy, E.H, M. Fleischman, and A. Philpot. 2003. The Omega Ontology. In prep.

McCallum, A.K. 1996. Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering. http://www-2.cs.cmu.edu/~mccallum/bow/.

Klavans, J.L. and S. Muresan. 2000. DEFINDER: Rule-Based Methods for the Extraction of Medical Terminology and their Associated Definitions from On-line Text. *Proceedings of AMIA Annual Symposium*, Los Angeles, CA.

Klavans, J.L. and B. Whitman. 2001. Extracting Taxonomic Relationships from On-Line Definitional Sources Using Lexing. *Proceedings of ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Roanoke, VA.

Klavans, J.L., P.T. Davis, and S. Popper. 2002. Building Large Ontologies using Web-Crawling and Glossary Analysis Techniques. *Proceedings of the NSF's dg.o 2002*. Los Angeles, CA.

Knight, K. and S.K. Luk. 1994. Building a Large-Scale Knowledge Base for Machine Translation. *Proc. of AAAI*.

Knoblock, C.A., K. Lerman, S. Minton, and I. Muslea, 2000. Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach. *Data Engineering Bulletin* 23(4).

Muresan, S., S. Popper, P. Davis, J.L. Klavans. 2003. Building a Terminological Database from Heterogeneous Definitional Sources. *Proceedings of the NSF's dg.o 2003*. Boston, MA.

Mahesh, K. and S. Nirenburg. 1995. A Situated Ontology for Practical NLP. *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, at IJCAI. Montreal, Canada.

Philpot, A., J.L. Ambite, and E. Hovy. 2002. DGRC AskCal: Natural Language Question Answering for Energy Time Series. *Proceedings of the NSF's National Conference on Digital Government dg.o 2002*. Los Angeles

Resnik, P. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D., UPenn.

Wilks, Y.A., B.M. Slator and L.M. Guthrie. 1996. *Electric Words: Dictionaries, Computers, and Meanings*. MIT.