In *Proceedings of the First National Conference on Digital Government (dg.o 2001).* Los Angeles, May 2001.

Data Acquisition and Integration in the DGRC's Energy Data Collection Project

Eduard Hovy*, Andrew Philpot*, Jose Luis Ambite*,
Yigal Arens*, Judith Klavans†, Walter Bourne†, and Deniz Saroz†

* Digital Government Research Center
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695
hovy@isi.edu
http://www.dgrc.org

† Digital Government Research Center
Department of Computer Science
Columbia University
535 West 114th Street, MC 1103
New York, NY 10027
klavans@cs.columbia.edu
http://www.dgrc.org

Abstract

The EDC project is developing new methods to make data that has been represented in disparate ways and stored in heterogeneous forms available to users in an integrated, manageable, and understandable way. Our approach is to represent the structure and types of data in the disparate collections in a standard format (called a domain model) and then to embed the domain model(s) into a large overarching taxonomy of terms (the ontology). Once thus standardized, the data collections can be found and retrieved via a single interface and access planning system. A major bottleneck, however, is the rapid inclusion of new data collections into this system. This paper describes some recent research in developing methods to automate the acquisition and inclusion processes.

1 Introduction

The massive amount of statistical and text data available from several Federal Agencies has created a daunting challenge for anyone wishing to use it. Two principal problems are apparent. On the one hand, the variety of terminology used is confusing: what one agency calls *salary* another might call *income*, and what a third also calls *income* might by its definition mean something rather different. On the other, the data is made available in a variety of forms, ranging from databases in Microsoft Access, to text documents with data arranged in columns, to webpages in html, to documents represented in pdf, etc.

In the Energy Data Collection (EDC) Project of the Digital Government Research Center (jointly at the University of Southern California's Information Sciences Institute and Columbia University's Computer Science Department), we have been addressing these and other problems of the past two years. We have been working with petroleum information obtained from the Federal Government's Energy Information Administration (EIA), the Bureau of Labor Statistics (BLS), the Census Bureau (Census), and the California Energy Commission (CEC). Typically, this information provides prices at various times for various types of gasoline. Our approach has been the following.

In order to achieve terminology standardization, we created a single model of the parameters that describe the data (the Domain Model). As described in Section 2, these parameters include:

- Types of gasoline: regular unleaded midgrade, etc.
- Units of volume measured: barrel, gallon, etc.

- Areas/regions in which prices apply: US, Adjustments-excluding-Alaska, California, Maine, etc.
- Time periods and frequencies in which prices were measured: annual, monthly, weekly, etc.

This Domain Model consists of approx. 500 nodes, and was built manually with the occasional assistance of domain experts from the Government agencies. In order to place it within a general context of other concepts (for users not particularly expert in gasoline), we embedded the Domain Model into our large concept taxonomy, the SENSUS Ontology (Section 3.1). This work required locating in SENSUS the nodes that appropriately represent concepts in the Domain Model, or generalizations (superconcepts) of them. Given that SENSUS contains 90,000 nodes, this is not an easy task. However, when new information is added to the Domain Model, for instance some hundreds or even thousands of new concepts and their variations, embedding them into SENSUS becomes too tedious to perform manually. In Section 3 we describe the work of trying to automate this process.

Once a Domain Model is built, we face the second problem: rapidly appropriating a new data collection, ‘wrapping’ it with code that interfaces with the particular form of implementation (pdf, webpages, database, etc.), and thereby enabling one to extract the desired data in a uniform way without worrying about each data collection’s native format. As described in Section 2, we manually created wrappers for several hundred data series and then semi-automatically wrapped approx. 58,000 more, from several agencies, and out of various sources. We used a commercial package to ‘unpack’ pdf format in some cases, interfaced with Microsoft Access in others, and wrote our own html parser in others.

All this work took place within the context of the EDC system (Ambite et al., 2001; Ambite et al., 2000).

2 Data Wrapping and Incorporation

2.1 Data Series Handled

To date, we have wrapped the following time series of gasoline sales:

BLS: 53 series, all of which are obtained by a wrapper which instantiates a form on the BLS web site.

- 27 of these are BLS/WP “PPI Revision Commodities”
- 12 of these are BLS/PP “PPI Revision Current Series”
- 8 of these are BLS/AP “CPI Average Price Data”
- 6 of these are BLS/CU “CPI All Urban Consumers”

Each BLS series provides 10 fields, including BLS-PERIOD (a string encoding quarterly, monthly, etc.), FOUR-DIGIT-YEAR, ORIGINAL-SERIES-NAME, SERIES-ID, SOURCE-TAG, SOURCE-URL, … and ultimately VALUE (a number providing the value of measurement).

CEC: 3 CEC series, obtained by wrapping a single page on the CEC web site, and containing 8 fields.

EIA PSM: 16 EIA PSM (Petroleum Supply Monthly) series. PSM is a period publication. We have converted a snapshot of the PSM in pdf format to html, then wrapped it to obtain the 8 annual and 8 monthly measurements. Each record contains 11 fields.

EIA OGIRS: 25,000 modeled EIA OGIRS (Oil and Gas Information Resource System) series. OGIRS is a data product available from EIA on CDROM; it is implemented as a query system on top of MS Access. We can access the Access data directly; also, we have imported the data themselves into Oracle. OGIRS has much formal metadata that can be viewed in Access, but retrieving it is not straightforward.

Each record contains 11 fields. Besides the time series concepts (the MEASUREMENTs), the EDC system also provides footnote information, by joining through parallel ANNOTATION and FOOTNOTE concepts, which share the definitional metadata above but have a few different retrievable attributes.

All this data is organized into a single Domain Model, which in turn is embedded into SENSUS. The Domain Model has the following primary dimensions (data 'columns').

The primary key for MEASUREMENT is {SERIES-ID STANDARD-DATE-NUMBER} which means that a SERIES-ID must be found or imposed for each source that encodes all identifying geographic, frequency, product, etc., information. Generally such IDs are available naturally in the source. Additionally, as an engineering practice, we have imposed 10 definitional attributes onto every domain concept. Each measurement concept thus possesses all of the following attributes, which can be retrieved as if they exist in an end source.

AGENCY-NAME	string	agency providing data
AREA-NAME	string	locale of data
FAMILY-NAME	string	subgroup of series within agency
FREQUENCY-NAME	string	how often/when measured, e.g. monthly
POINT-OF-SALE-NAME	string	where in the supply chain measured
PRODUCT-NAME	string	what product e.g. unleaded regular
PROVENANCE-NAME	string	how data obtained: web, RDBMS, etc.
QUALITY-NAME	string	what kind of measurement: vol, price
SEASONAL-ADJUSTMENT-NAME	string	is data seasonally adjusted
UNIT-NAME	string	units of measurement: Mbbl/mo, etc.

2.2 Wrapping Webpages for EIA

Recently, at the request of EIA, we wrapped some 15,000 location-specific series from a collection of data tables defined as text. The information had been cleared for publication in a format that listed a certain gasoline sales values for each state, all combined in one table (e.g., http://www.eia.doe.gov/emeu/states/ main_ca.html). In order to publish all monthly values for each state individually, the original webpages had to be wrapped, merged into a single large collection, and then reaggregated as required by appropriate extraction (e.g., http://www.eia.doe.gov/pub/oil_gas/petroleum/data_publications/petroleum_marketing_monthly/current/txt/tables31.txt).

Since this data changes monthly, a natural approach here was to generate snapshot breakout reports of each desired composite report, and publish them to the web each month.

The webpages are structured into four different table formats. Using ISI's existing wrapper technology, two of these formats, comprising some 14,500 series (95% of the data), were wrapped, merged, and reaggregated within a few days. The final two formats were handled in a subsequent week.

In detail, this process required three steps:

1. Wrapping the data. By hand, we classified each text report file's structure into one of three forms. We then used ISI's Ariadne (Knoblock et al., 2001) table interpretation wrapping technology (Knoblock et al., 2000) to extract a series of pages from each multi-page report according to the (form-specific) characteristic set of text landmarks and text entries. We identified left- and right- hand pages when present, then broke each page into header, body, and footer (with footnotes). Headers were wrapped to extract type/subtype and measurement/column labels. Bodies were wrapped to extract information in columns, using landmarks, normal text conventions, and apparent column widths. Table conventions, landmarks, and limited amount of domain-specific information were used to repair multi-line tokens and to associate subtotals with their superheadings. The resultant relation was saved in a neutral interchange format. For example, http://edc.isi.edu/textwrap/example/form1/monthly/tables31.txt.csv shows the original tables "flattened" into 7-tuples (product,frequency,locale,date,subtype,measurement,value).
2. Posing appropriate queries against the wrapped data. An EIA representative specified requirements for per-state breakouts from composite reports, which effectively require database selections, projections, and potentially unions between the relations. In a completed system, data in the interchange format from step

1 above would be imported into a RDBMS or integrated by an information source mediator such as SIMS. At this point, we implemented a simple emulation that supports only conjunctions of selections and projections.

3. Html re-presentation of the results. This is a more open-ended process than the above. At present we are re-integrating them as an html rendition of the original headings, effectively to splice the original heading together with only the rows relevant to a state-based breakout. We honor the original heading/subhead as much as possible, although subhead totals may no longer be relevant. Finally, we hope to parameterize the output using html Cascading Style Sheets instead of significant embedded style tags, to reduce file size and support future malleability of styles. Initial results can be seen at http://edc.isi.edu/textwrap/example/form1/monthly/tables31.txt__California_ Regular__monthly.html.

2.3 Future Work: Extensions Planned

The particular data we extracted (Petroleum Marketing Monthly/Annual) are both consistent with and complementary to the other EIA data we have used (OGIRS and PSM). Therefore, we can retarget this data into our current and ongoing Energy Data Collection application.

At the moment, we are not using Ariadne to extract footnotes from the page footer subdocuments. This is only slightly more involved than the above. In EDC, we have a modeling convention wherein we use three model-level overlays on each modeled source, allowing us to associate an arbitrary number of independent footnotes to tables, columns, rows, and individual cells; this modeling framework can be applied directly to this data if we generate text wrappers for each form type's footnote reports. Currently, each footnoted measurement is generated as a tuple of [tag,value], but it would be straightforward to abstract this out into individual raw data, footnote tag, and footnote attachment reports (as we did for the EIA PSM and BLS web data for EDC).

General geometric reasoning to understand multi-column headings (e.g., http://edc.isi.edu/textwrap/example/form1/monthly/tables31.txt__page__001__head) is not currently a part of Ariadne's reasoning process. We plan to investigate certain approaches in the table understanding and induction community.

3 Toward Automating Domain Modeling

3.1 SENSUS Ontology

As described above, we embed the Domain Model (the parameters defining the data collections) into the SENSUS ontology, a 90,000-node taxonomy of terms covering most areas of human expertise (Knight and Luk, 1994). SENSUS, built at USC/ISI out of WordNet (from Princeton University) and ISI's Penman Upper Model, links terms together into a subsumption (*isa*) network, with additional links for *part-of*, *pertains-to*, and so on.

Our work focused on extending SENSUS to incorporate new domain models, and extending and developing new automated concept-to-ontology alignment algorithms. We used a set of 6,000 terms, extracted automatically by team members from Columbia University from glossaries provided by EIA, Census SICS and NAICS codes, and EPA.

3.2 Semi-Automated Data and Domain Modeling

We would like to help automate the process of creating Domain Models. Our approach is to use machine learning and data mining techniques to identify important characteristics of the data source in order to be able to recognize what kind(s) of data are included. This very ambitious (and hence long-term) goal involves breaking down the work as follows. Given some new domain:

- Text (glossaries, manuals, etc. associated with the data): extract information from the text, using Finite State and other techniques (Klavans and Muresan, 2000); create formally defined concepts

- Databases: create concepts out of the metadata
- Domain Model: (manually) use this information to build a domain model
- Ontology matching: cluster and embed these concepts into SENSUS (Hovy, 1998)

The following provides more detail.

1. Text Analysis/Information Extraction (from glossary definition to a structured template): Columbia's finite-state methods employ templates to identify standardized items such as dates, dollar amounts, etc. (see http://www.cs.columbia.edu/nlp/flkb/eia_large/), while ISI's parsers identify complex NPs and VPs.

2. Database Concept Extraction: This process is still under development. We have outlined a procedure, similar to work is being done in the data mining community (e.g., (Doan et al., 2000). Input sources contain data (numeric; string); metadata (table and column names; text (including footnotes, glossaries, and text); Ontologies (terminological (Sensus) or logical (CYC)); and partial Domain Models. Possible characterizations of a single attribute include ranges, enumerated sets, minimum, maximum, average values, orthographic patterns (e.g., (xxx) xxx-xxxx), format typing (numbers, letters). Atoms include cell values, attribute names, table names, and source names.

3. Domain Modeling: This task is still performed manually.

4. Ontology Matching: We are developing a sequence of steps to help automatically link one or more concepts (e.g., a Domain Model) into SENSUS. First, for a given (set of) concepts, propose candidate equivalents or near-equivalents in SENSUS (see match algorithms in Section 3.3). Then, given the set of candidate matches, rank them and select the best one(s) (see Dispersal Match in Section 3.3).

3.3 Clustering and Alignment Research, and Glossary Linkage

Columbia University has extracted approx. 6,000 concepts from glossaries and other text acquired from EIA and BLS. Our challenge was to link these concepts into SENSUS at their most appropriate points. We developed the process shown in Figure 1, with times required to handle the 6000 glossary concepts (this work is almost complete, but still needs evaluation).

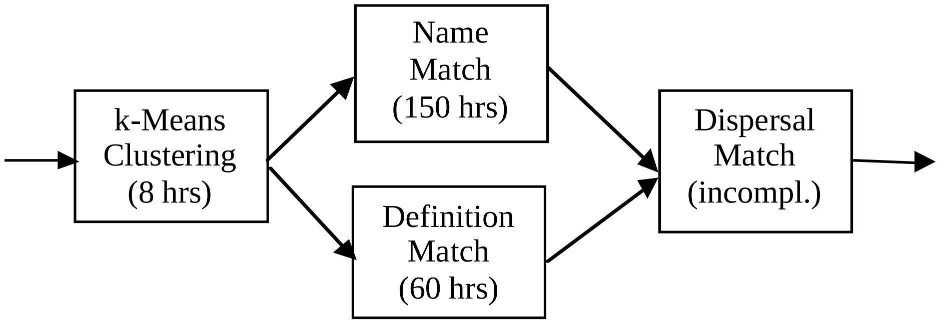

Figure 1. Concept-to-ontology alignment. Times are to embed 6,000 concepts.

1. Prepare the concepts to be linked: format them appropriately, stem definitions, etc.
2. Cluster the concepts into semantically related clusters of between 20 and 50 concepts each. For this we tested various clustering algorithms and implemented a fast version of k-Means (see below).
3. Per cluster, apply the Name and Definition Match algorithms (Hovy, 1998) to each concept, and collect for each concept the candidate matches in SENSUS (see below).
4. Apply the Dispersal Match algorithm (newly invented) to each cluster, in order to select from each set of candidate matches the most appropriate one for each concept. Return for each concept a ranked list of alignments (see below).
5. Evaluate the alignments manually, and improve them as required.

Clustering:

Definition: for a given set of concepts, cluster them to find the sets of ones that belong together semantically. One of three criteria has to be provided by the user in determining cluster cutoff:

- number of clusters desired
- similarity cutoff value
- max/min size of desired clusters

In addition, the user has to specify how the inter-subcluster distance is measured; variations include

- slink, clink, median, and Ward's method
- speedup: recursive nearest-neighbor (RNN)

usually using a Euclidean distance (similarity) metric, defined in a vector space of word tokens.

We tried the above clustering methods (using ISI's clustering package ISICL), after having reimplemented some of them to work more efficiently and to link to a GUI that would display the results graphically. We then implemented a version of k-Means clustering for text and numeric vectors, with both Euclidean and spherical distance measures.

Name Match:

Definition: match the name of the concept against the name of each ontology concept and rank the goodness of fit.

Variations:

- longest common substring (contiguous)
- position of match: equal beginnings, ends, or wholly contained
- prefix/suffix structure: *leaded* vs *unleaded*, *happy* vs *happiness*
- stemming: root words or inflected words
- non-word matches: letter trigram match
- case sensitivity

Name match includes a substring match operation, which in our first implementation was extremely slow. We later implemented one version of an algorithm used in molecular biology to match DNA subsequences. The results were still daunting—over 150 hours to match 6,000 concept names against all of SENSUS. For this reason we implemented a match of just letter trigrams (breaking the name into all its contiguous 3-letter units), which took approx. 24 hours but gave less satisfactory results.

Definition Match:

Definition: match the English definition of the concept against the English definition of each ontology concept and rank the goodness of fit. Variations:

- text to match: word tokens or letter trigrams
- stemming: root forms or fully inflected
- include or exclude stop words
- terms to match: basic definition vs. expanded definitions (SENSUS synonym sets)
- text to match: core definition vs. examples included
- IR similarity measure in a word vector space: Dice coefficient, cosine, Jaccard, etc.
- word weighting function: none vs. *tf.idf*

Dispersal Match:

Definition: for a (semantically related) cluster of concepts, consider the total set of candidate matches in the ontology. By picking one candidate for each concept, find the smallest (tightest, therefore most closely related semantically) cluster of candidates in the ontology. The variations include:

- full algorithm (combinatoric complexity) vs. greedy search
- number of candidates allowed per concept

- relative weighting of candidates from Name and Definition Matches

This match was invented for this project, and tested on a small set of 100 concepts, 10 candidates each, early in 2000. It performed very well (Ambite et al., 2001). However, given the current number (6,000) of concepts to be matched, we could not implement the full combinatoric algorithm. We therefore used a greedy search approach. Unfortunately, our experiments with this version of the Dispersal Match gave far less encouraging results. We therefore decided to test the performance of the whole sequence on ideal data, in order to determine where the problems arose.

To create the ideal data, we extracted from the ontology three subhierarchies (furniture, beverages, and tools). We applied the match sequence to these hierarchies in order to see how well they were matched to their true locations. We expected that the Dispersal Match would find the best matches easily, but this turned out to be not the case. The confusion matrix in Table 1 shows that the results, though much better than random, are in some cases rather poor. While for example Cluster 3 is 95.76% pure Beverages (Column distribution, bottom row of table), Cluster 1 is equally split among the three alternatives. Cluster 2 has swallowed most of both Tools and Furniture (middle row, Row distribution).

	Tools	**Beverages**	**Furniture**	**Totals**
Global % distribution				
Cluster 1	5.90%	5.76%	6.91%	18.56%
Cluster 2	35.83%	4.03%	17.84%	57.70%
Cluster 3	0.72%	22.73%	0.29%	23.74%
Totals	42.45%	32.52%	25.04%	100.00%
Row distribution				
Cluster 1	13.90%	17.70%	27.59%	
Cluster 2	84.41%	12.39%	71.26%	
Cluster 3	1.69%	69.91%	1.15%	
Totals	100.00%	100.00%	100.00%	
Column distribution				
Cluster 1	31.78%	31.01%	37.21%	100.00%
Cluster 2	62.09%	6.98%	30.92%	100.00%
Cluster 3	3.03%	95.76%	1.21%	100.00%

Table 1: Confusion matrix showing results of concept matching on three sets of ideal data. Ideally, one cell in each row (or column) will contain 100%, and the others in that row (or column) 0%.

One reason for the disappointing results was inadequate matching, which produced spurious candidate matches. For this reason, we decided to upgrade SENSUS to WordNet 1.6, which includes both a more logical internal organization and better and more complete concept definitions. At the time of writing we have completed the transition to WordNet 1.6. The other reason is inaccurate clustering.

4 Conclusion

As we encounter more data and new domains, the need grows for efficient and automated procedures for wrapping new data collections, inducing their Domain Models by reference to associated text, the ontology, similar domain models, etc.), and embedding everything into the Ontology. We plan to continue investigating algorithms on these topics.

References

Ambite, J.L., Y. Arens, E.H. Hovy, A. Philpot, L. Gravano, V. Hatzivassiloglou, J.L. Klavans. 2001. Simplifying Data Access: The Energy Data Collection Project. *IEEE Computer* 34(2).

Ambite, J.L., Y. Arens, L. Gravano, V. Hatzivassiloglou, E.H. Hovy, J.L. Klavans, A. Philpot, U. Ramachandran, J. Sandhaus, A. Singla, and B. Whitman. 2000. Building Ontologies and Integrating Data from Multiple Agencies: A Case Study Using Gasoline. Paper 1941, *2000 Joint Statistical Meetings*. Indianapolis, IN.

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).

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).

Hovy, E.H. 1998. Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. *Proceedings of the 1st International Conference on Language Resources and Evaluation* (LREC). Granada, Spain.

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 American Medical Informatics Association (AMIA) Annual Symposium*, Los Angeles, CA.

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

Knoblock, C.A., S. Minton, J.L. Ambite, N. Ashish. I. Muslea, A.G. Philpot, and S. Tejada. 2001. The Ariadne Approach to Web-based Information Integration. *International Journal on Cooperative Information Systems* (IJCIS) 10(1–2) Special Issue on Intelligent Information Agents: Theory and Applications (145–169).

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).