Aligning Database Columns using Mutual Information

Patrick Pantel, Andrew Philpot and Eduard Hovy
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
{pantel,philpot,hovy}@isi.edu

ABSTRACT

As with many large organizations, the Government's data is split in many different ways and is collected at different times by different people. The resulting massive data heterogeneity means government staff cannot effectively locate, share, or compare data across sources, let alone achieve computational data interoperability. A case in point is the California Air Resources Board (CARB), which is faced with the challenge of integrating the emissions inventory databases belonging to California's 35 air quality management districts to create a state inventory. This inventory must be submitted annually to the US EPA which, in turn, must perform quality assurance tests on these inventories and integrate them into a national emissions inventory for use in tracking the effects of national air quality policies. The premise of our research is that it is possible to significantly reduce the amount of manual labor required in database wrapping and integration by automatically learning mappings in the data. In this research, we applied statistical algorithms to discover correspondences across comparable datasets. We have seen particular success in an information theoretic model, called *SIfT* (Significance Information for Translation), that performs data-driven column alignments. We have applied *SIfT* to mapping the Santa Barbara County Air Pollution Control District's 2001 emissions inventory database with the California Air Resources Board statewide inventory database. A fully customizable interface to the *SIfT* toolkit is available at http://sift.isi.edu/, allowing users to create new alignments, navigate the information theoretic model, and inspect alignment decisions. On a broader scale, this work makes strides toward appeasing a central problem in data management of integrating legacy data.

Categories and Subject Descriptors

H.2.5 [**Database Management**]: Heterogeneous Databases.

General Terms

Algorithms, Experimentation.

Keywords

Information theory, mutual information, database alignment.

1. INTRODUCTION

Due to the wide range of geographic scales and complex tasks that the Government must administer, its data is split in many different ways and is collected at different times by different agencies. The resulting massive data heterogeneity means one cannot effectively locate, share, or compare data across sources, let alone achieve computational data interoperability.

To date, all approaches to wrap data collections, or even to create mappings across comparable datasets, require manual effort. Despite some promising work, the automated creation of such mappings is still in its infancy, since equivalences and differences manifest themselves at all levels, from individual data values through metadata to the explanatory text surrounding the data collection as a whole. More general methods are required to effectively address this problem.

Viewing the data mapping problem as a variant of the cross-language mapping problem in Machine Translation (MT), we employed new statistical text alignment and clustering algorithms developed in Natural Language Processing to discover correspondences across comparable datasets. In this paper, we present an information theoretic model, which we call *SIfT* (Significance Information for Translation), that performs data-driven column alignment. The key to our approach is to first identify, using the information-theoretic model, the most informative data elements and then match data sources that share these informative elements. This work has the potential to significantly reduce the amount of human work involved in creating single-point access to multiple heterogeneous databases.

2. GOVERNMENT COLLABORATION

We are working with the following set of domain data. Emissions inventories are being provided by staff at the California Air Resources Board (CARB) in Sacramento, who are faced with the challenge of integrating the emissions inventory databases belonging to California's 35 Air Quality Management Districts (AQMD) to create a state inventory. This inventory must be submitted annually to the US EPA which, in turn, must perform quality assurance tests on these inventories and integrate them into a national emissions inventory.

To deliver their annual emissions data submittal to CARB, air districts have to manually reformat their data according to the specifications of CARB's emission inventory database called California Emission Inventory Development and Reporting System (CEIDARS). Every time the CEIDARS data dictionary is revised (as has happened several times recently, for example in

2002), work is required on the part of AQMD staff to translate emissions data into the new format. Likewise, when CARB provides emissions data to US EPA's National Emission Inventory (NEI), significant effort is required by CARB staff to translate data into the required format.

Our development testbed is the 2001 Santa Barbara County Air Pollution Control District (SBCAPCD) emissions inventory, one of the 35 California air districts. Locally, SBCAPCD's database contains 20 tables with approximately 300 columns, 160,000 rows, and 1.8 million data elements. The target CEIDARS schema contains only 6 tables with a total of 260 columns, 70,000 rows, and 2.4 million values.

We expect that *SIfT* will greatly reduce the manual effort required to build the wrappers and mappings required for all the entirety of California.

3. RELEVANT WORK IN ALIGNMENT

Although there is a wealth of information in government databases, a lack of standardization has made it very difficult to integrate various data sources. Integration and reconciliation of data across non-homogeneous databases is an old but unsolved and ever-growing problem. Some mechanism is required to standardize data types, reconcile slightly different views, and enable sharing. For textual data, the information retrieval approach exemplified in web search engines such as Google and Yahoo! works reasonably well to find exact and close matches (around 40% recall and precision, over the past decade, determined at the annual TREC[1] conferences).

For numerical data, however, search engines are inappropriate. Instead, two approaches are possible. Either one can build a central data model that integrates the specialized metadata for each database, or one can create direct mappings across the data (cells, columns, rows, etc.) of the databases themselves. Both approaches are difficult. With regard to the former, various methods have been developed. The "global-as" method [2][3] assumes the central model is complete, but that local databases may deviate from it; access is via the central model. This model requires serious effort to extend. In contrast the "local-as" method [8] assumes that the central model is incomplete, simply narrowing the sources to be further searched, which may require tedious additional search effort. In contrast, the "ontology method" uses a single overarching super-metadata model (the ontology) into which all databases' metadata descriptions are subordinated hierarchically [1][6].

The second general approach, creating mappings across individual (subsets of) data, is impossible to bring about for real-sized data collections, unless automated methods are used to find the mappings. Machine learning methods are only now being applied to this problem. Schema-based matching algorithms [12] align databases by matching the meta-data available in the databases (e.g., two tables with column name *zip_code* are aligned; most approaches will also match columns labeled *zip_code* and *zip*). However, since there is often no standardized naming scheme for meta-data, schema-based sometimes fails. Instance-based matching algorithms align databases using the actual data [5]. These methods typically fail when different columns share a common domain (e.g., business vs. residence phone numbers) or when matching columns that exhibit different encodings (e.g., a phone number field stored as a text string in one database and stored as a numerical field in another). Kang and Naughton [7] propose an information-theoretic model to match unaligned columns after schema- and instance-based matching fails. Given two columns *A.x* and *B.x* that are aligned, the model computes the association strength between column *A.x* with each other column in *A* and column *B.x* and each other column in B. The assumption is that the highly associated columns from *A* and *B* are the best candidates for alignment. In this paper, we adopt a similar information-theoretic model for instance-based matching.

[1] The Text REtrieval Conference (TREC) provides the infrastructure necessary for large-scale evaluation of text within the information retrieval community.

4. DATA-DRIVEN ALIGNMENT

The key to our approach is to first identify, using an information-theoretic model, the most informative data elements and then match data sources that share these informative elements. For example, in our case study of matching SBCAPCD and CARB schemas, since the source data is from Santa Barbara County, we expect that many of the columns in SBCAPCD will contain the word "Santa Barbara" (e.g., factory names, locations, addresses, etc.) However, only one column contains the word "Wingerden." Therefore, a random pair of columns from SBCAPCD and CARB that both contain the data element "Santa Barbara" are intuitively not as similar as if both columns contained the data element "Wingerden." Our model will automatically detect that "Santa Barbara" is less informative than "Wingerden" and will consequently assign a higher similarity to two columns that share only "Wingerden" rather than only "Santa Barbara".

4.1 Information Theoretic Model

Informative elements are modeled in *SIfT* using an information theoretic measure called mutual information. Similar columns are discovered using a clustering algorithm called CBC [9].

In any clustering application, the critical step is representing the data such that elements group together according to our desired output. For example, if we want to cluster medical patients according to their possible diseases, we might represent them by their height, weight, age, gender, whether they smoke or not, etc.; we would not, however, represent them by their favorite board game or favorite movie since with this representation we would likely group patients according to their entertainment preferences.

The representation of an element is often called a feature vector (or vector space model). Each feature is simply a measurement of the element. For example, in clustering data points on a 3-dimentional graph, we would represent each point using three features: the *x*, *y*, and *z* coordinates. These three measurements completely describe the points.

4.1.1 Feature Representation

In aligning inter-database columns *s* and *t*, we assume that *s* and *t* contain similar but not necessarily identical fields (accounting for noise and discrepancies in the data). One representation for columns is simply the data fields they contain. Consider the following database columns taken from two databases *S* and *A*:

```
S.phone.number:
     310-555-6789, 310-555-0987,
     780-433-9393, …
A.area:
     310, 310, 780, …
A.ph:
     555-6789, 555-0987, 433-9393, …
```

We could represent these columns using their field values with a frequency of occurrence as measurement. For the above example, the feature vectors using this representation would be:

```
S.phone.number:
     310-555-6789      1
     310-555-0987      1
     780-433-9393      1
A.area:
     310               2
     780               1
A.ph:
     555-6789          1
     555-0987          1
     433-9393          1
```

Notice that none of these features overlap and consequently a clustering algorithm would not discover any similarity between the columns. In this research, we enrich the feature space by classifying data columns within several feature domains (e.g., zip code, phone number, state, positive integer). Once a column is classified within a particular feature domain, the feature types associated with that domain are extracted for the column's feature vector (e.g., *zip5* – the first five digits of a zip code, *zip4* – the last four digits of a nine-digit zip code, *area* – the area code of a phone number, *exch* – the 3-digit phone number exchange, *phone* – the seven-digit local phone number, *ext* – the extension of any digits after a 10-digit phone number). We also add domain specific feature domains. For example, in aligning the EPA Air Quality databases, we added feature domains for SIC codes (OSHA), NAICS codes (CENSUS), EIC codes (CARB), SCC point source codes (EPA), and CAS registry numbers (EPA2).

The algorithm we use for recognizing these domains simply searches for patterns that describe the domain. For example, a 10-digit phone number is recognized if the first three digits are a known area code, the fourth digit is between [2-9], and the rest of the field is numeric. If our patterns do not fire on a particular column (e.g., a column containing international phone numbers), then the catch-all Text feature domain will always fire.

We allow the user of the system to decide which feature domains and associated feature types are active for any given alignment. Suppose a column is identified as a phone number and we decide to extract feature types *area* and *phone* for all phone numbers. Then for each field such as "310-555-6789", the system extracts two features with frequency 1:

```
area:310                  1
phone:555-6789            1
```

Similarly, for fields such as "555-6789", we extract a single feature:

```
phone:555-6789            1
```

Now, we see some overlap between the columns *S.phone.number* and *A.ph* from the previous section. A clustering algorithm could therefore discover a similarity between the two columns.

4.1.2 Mutual-Information Vector-Space Model

Representing data for clustering requires both a feature representation and a measurement of the features. We now describe our model for measuring the feature types described in the previous section.

Above, we measured each feature by its frequency of occurrence. However, certain features are more informative than others. For example, the common word '*the*' will be present in many text strings. Two strings that happen to contain the word '*the*' does not indicate as much similarity as if they contained an uncommon word such as 'carbon'.

Pointwise mutual information is commonly used to measure the association strength between two events [4]. It essentially measures the amount of information one event gives about another. For example, knowing that a column contains the word '*the*' is not informative of the contents of that column (because *the* is common across many columns). Conversely, if very few columns contain the word *carbon*, then that word is an informative feature (i.e. if columns *p* and *q* from different databases happen to contain *carbon*, then they are more likely to be aligned than if they shared the word *the*).

The pointwise mutual information between two events x and y is given by:

$$mi(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

Mutual information is high when x and y occur together more often than by chance. Mutual information compares two models for predicting the co-occurrence of x and y: one is the MLE (maximum-likelihood estimation) of the joint probability of x and y and the other is some baseline model. In the above formula, the baseline model assumes that x and y are independent. Note that in information theory, mutual information refers to the mutual information between two random variables rather than between two events as used in this paper. The mutual information between two random variables X and Y is given by:

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

The mutual information between two random variables is the weighted average (expectation) of the pointwise mutual information between all possible combinations of events of the two variables.

For each element e, we first construct a frequency count vector $C(e) = (c_{e1}, c_{e2}, \ldots, c_{em})$, where m is the total number of features and c_{ef} is the frequency count of feature f occurring in element e. Here, c_{ef} is the number of times column e contained a feature f. For example, in column e = *A.area* from Section 4.1.1, one feature is *area:310* with count 2.

We then construct a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$ for each column e, where mi_{ef} is the pointwise mutual information between column e and feature f, which is defined as:

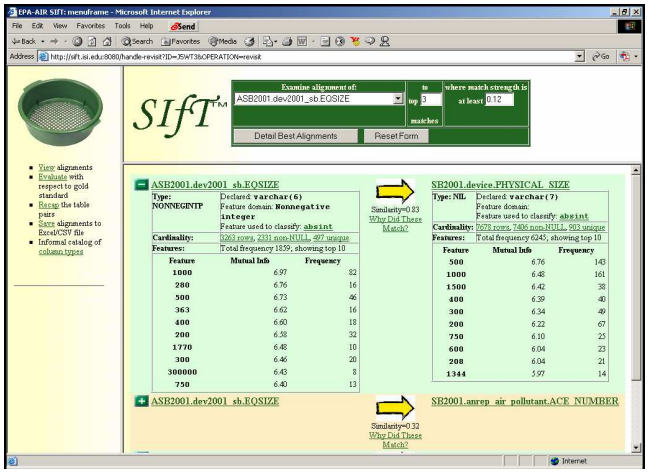

Figure 1. A correct alignment discovered by *SIfT* for the *EQSIZE* column in the CARB database to the *Physical Size* column in the SBCAPCD database.

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{i=1}^{m} c_{ej}}{N}}$$

where n is the number of columns and $N = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}$ is the total frequency count of all features of all columns.

A well-known problem is that mutual information is biased towards infrequent elements/features. We therefore multiply mi_{ef} with the following discounting factor:

$$\frac{c_{ef}}{c_{ef}+1} \times \frac{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)}{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)+1}$$

4.2 Similarity Metric

To cluster elements, we need a measure of similarity (or distance) between them. We construct a matrix containing the similarity between each pair of columns e_i and e_j using the cosine coefficient of their mutual information vectors [10]:

$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i,f} \times mi_{e_j,f}}{\sqrt{\sum_f mi_{e_i,f}^{\ 2} \times \sum_f mi_{e_j,f}^{\ 2}}}$$

This measures the cosine of the angle between two mutual information vectors. A similarity of 0 indicates orthogonal vectors whereas a similarity of 1 indicates identical vectors. For two very similar elements, their vectors will be very close and the cosine of their angle will approach 1. A nice property of the cosine metric is that it is not very sensitive to 0 frequency features. Hence, given a column containing phone numbers of all customers/stores and another column containing only customer phone numbers, cosine will find a similarity even though all store phone numbers will have frequency 0 in the second column. In other words, the absence of a matching feature is not as strong an indicator of dissimilarity as the presence of one is an indicator of similarity. Other measures like the Euclidean distance do not make this distinction.

4.3 Alignment

SIfT uses the similarity matrix generated by the clustering algorithm. For each column from source database *A*, we align it with its most similar columns from target database *S* such that the similarity is above a certain threshold θ.

5. EVALUATION

Preliminary inspections using *SIfT* show very promising results. The system was able to find 75.0% of the mappings that were manually inserted into a gold standard. *SIfT* also found several mappings that were undetected by a human annotator.

Given databases *A* and *B*, the ideal goal of our task is to automatically generate the same column alignment that a human expert would generate. A step forward is to greatly reduce the number of alignment pairs to consider by a human expert. We evaluate our system by measuring its precision and recall, and by measuring its reduction in human effort to generate a manual alignment.

5.1 Data Set

The source material we use for mappings, in the form of individual data sets, metadata schemas, etc., was provided by the Santa Barbara County Air Pollution Control District (SBCAPCD). SBCAPCD provided a complete archive of the emissions inventory it conducted for 2001 and 2002, covering facilities, devices, processes, permitting history, as well as criteria and toxic emissions. Mapping target material, including an integrated database and its metadata schema, was provided by the California Air Resources Board (CARB), in the form of the statewide emissions inventory for 2001 and 2002. Negotiations with other air districts are also underway.

For the purposes of our evaluation, we focused on the 2001 SBCAPCD and 2001 CARB data. The SBCAPCD and the CARB emissions inventory databases used in our experiments each contain approximately 300 columns, thus a completely naïve human must consider 90,000 alignment decisions.

5.2 Gold Standard

We conducted a hand-alignment of the Santa Barbara County APCD 2001 emissions data to the structure and contents of the California ARB emissions inventory, limiting ourselves to the portion of the ARB data set covering SBC APCD. We broke the task into two main subtasks: (1) Intra-source alignment, or discovering and documenting internal relationships, such as foreign keys, within the two components; (2) Inter-source alignment, or aligning important key columns of SBC APCD with the best candidate columns of ARB, focusing on columns that did not appear to be used primarily for intra-source relations. As we were hoping to gain insight into the process which would further our data-driven approach, we did not make significant use of available metadata and background documentation. As we are not domain experts, and we did not take advantage of significant computational aids beyond simple spreadsheet-level bookkeeping, each of the two main tasks described above required roughly 40 hours of detail-oriented work by a professional research

Table 1. Precision, recall and F-measure of different feature representations.

	Precision		Recall		F-measure	
System	Micro	Macro	Micro	Macro	Micro	Macro
Simple	75.0%	65.0%	72.2%	65.0%	73.6%	65.0%
Trigrams	44.4%	44.4%	81.5%	80.0%	57.5%	57.1%
Rich	62.5%	57.7%	79.6%	75.0%	70.0%	65.2%

Table 2. Top-5 precision of different feature representations

System	Top-1	Top-2	Top-3	Top-4	Top-5
Simple	71.4%	92.9%	92.9%	92.9%	92.9%
Trigrams	66.7%	83.3%	83.3%	83.3%	83.3%
Rich	62.5%	93.8%	93.8%	93.8%	93.8%

programmer. This "gold standard" alignment, including annotations, was communicated back to the participating agencies. In a few cases these "noisy" match transforms were found to correspond to bugs in the data management work pattern within an agency.

The methodology for constructing a "gold standard" alignment was informal and *ad hoc*. It would be preferable to have a column-to-column alignment catalog agreed upon by the two agencies, but this was not available as it would require a potentially large investment of labor on the part of our local government partners to develop. The entirety of the gold standard, including annotations, is available from the *SIfT* url: http://sift.isi.edu/.

5.3 Precision and Recall

SIfT outputs for each column in a source database the columns to which it aligns in the target database. Figure 1 illustrates a screenshot of a correct alignment discovered by *SIfT* from CARB's *EQSIZE* column to SBCAPCD's *Physical Size* column.

The precision of the system is the percentage of correct alignment decisions:

$$Precision = \frac{C}{T_A}$$

where C is the number of correct alignment pairs and T_A is the total number of alignment pairs in the system alignment. This type of precision is often called micro-precision. Another precision, called macro-precision, averages the average precision of each column that is being aligned.

The recall of the system is the percentage of gold standard alignment pairs, T_G, that were retrieved by the system:

$$Recall = \frac{C}{T_G}$$

Precision and recall measure the tradeoff between identifying alignments correctly and getting all the possible alignment. For example, a system that returns all possible alignment pairs would achieve a recall of 100% but with an abysmal precision. Increasing the threshold θ from Section 4.3 increases the recall of the system but decreases the precision.

It is sometimes useful to have a single measure that combines precision and recall aspects. One such measure is the *F*-measure [11], which is the harmonic mean of recall and precision:

$$F = \frac{1}{\alpha\frac{1}{R} + (1-\alpha)\frac{1}{P}}$$

where R is the recall and P is the precision. Typically, $\alpha = ½$ is used:

$$F = \frac{2RP}{R+P}$$

F weighs low values of precision and recall more heavily than higher values. It is high when both precision and recall are high.

Table 1 shows the results comparing the precision, recall, and *F*-measure of various different feature representations. The Simple system simply represents each column by the fields it contains. The Trigram representation extracts letter trigram features for each field whereas the Rich representation extracts all possible features domains and feature types described in Section 4.1.1. Each representation uses the information-theoretic vector space model presented in Section 4.1.2. The *F*-measure of the simple representation is as good as if not better than the other two. This is due to the power of the mutual-information vector-space model which in effect automatically discovers the key values of a particular data domain.

Table 2 shows the precision of our system where the precision indicates the percentage of columns that have at least one correct alignment in the top-5 alignments. Interestingly, if the system can find a correct alignment for a given column, then the alignment will be found in the first two returned candidate alignments.

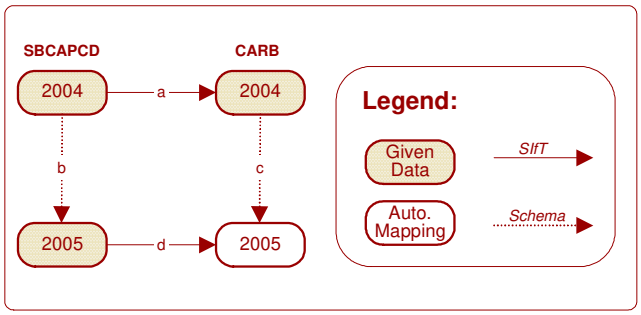

Figure 2. Closing the loop: the design for automatically generating a data transfer between SBCAPCD and CARB for 2005 given the 2004 data.

Inspecting only two candidate alignments for each possible column will greatly reduce the number of possible decisions made by a human expert.

6. DISCUSSION AND FUTURE WORK

In our experiments with SBCAPCD and CARB, each database contains approximately 300 columns. Thus, there are 90,000 alignment decisions that must be considered. Assuming that each alignment must be considered (in practice, many alignments are easily rejected by human experts) and that for each column we output at most k alignments, then a human expert would have to inspect only $k \times 300$ alignments. For $k = 2$, only 0.67% of the possible alignment decisions must be inspected, an enormous saving in time.

We are currently deploying *SIfT* on data from other California districts like Ventura County APCD and San Diego County APCD. However, our challenge lies not in the post-analysis of a data transfer between district and state in past years, but instead on linking new data as it becomes available each year. This is a challenge since the data formats may change on both sides (the collectors and the integrators). Since, however, changes year by year are not likely to be large, we can try to reconcile the possibly divergent evolutions automatically, thereby closing the loop by automatically generating the data transfers. Figure 2 shows our design for automatically generating a data transfer between a California district and CARB for 2005 given the 2004 data. First, applying *SIfT* as a post-analysis to the 2004 transfer (arrow *a*), we learn the mappings between the data sources for 2004 (this is what is presented in this paper). Then, given the schema changes for both SBCAPCD and CARB for 2005 (arrows *b* and *c*), we determine which mappings from *a* still hold. Human experts need only manually align those data elements from the (usually small) schema changes. Achieving this challenge, we hope that *SIfT* will be useful to our Government partners.

7. CONCLUSIONS

We proposed an information theoretic model, called *SIfT*, for performing data-driven column alignments. We have applied *SIfT* to the task of aligning the Santa Barbara County Air Pollution Control District's (SBCAPCD) 2001 emissions inventory database with the California Air Resources Board statewide inventory database. A fully customizable interface to the *SIfT* toolkit is available at http://sift.isi.edu/, allowing users to create new alignments, navigate the information theoretic model, and inspect alignment decisions.

This work has the potential to significantly reduce the amount of human work involved in creating single-point access to multiple heterogeneous databases. This problem is faced by thousands of large enterprises with numerous data collections, from Government agencies at all levels to the chemical and automotive industries to startup companies that link together and integrate websites. By automatically postulating mappings across databases/metadata, our algorithms will enable the database wrapper builder (whether fully manual or semi-automated) to work more quickly and effectively. It will also help with the creation of metadata standards.

8. REFERENCES

[1] Ambite, J.L.; Arens, Y.; Gravano, L.; Hatzivassiloglou, V.; Hovy, E.H.; Klavans, J.L.; Philpot, A.; Ramachandran, U.; Ross, K.; Sandhaus, J.; Sarioz, D.; Singla, A.; and Whitman, B. 2002. Data Integration and Access: The Digital Government Research Center's Energy Data Collection (EDC) Project. In W. McIver and A.K. Elmagarmid (eds), *Advances in Digital Government*. pp. 85–106. Dordrecht: Kluwer.

[2] Baru, C.; Gupta, A.; Ludaescher, B.; Marciano, R.; Papakonstantinou, Y.; and Velikhov, P. 1999. XML-Based Information Mediation with MIX. In *Proceedings of Exhibitions Program of ACM SIGMOD International Conference on Management of Data*.

[3] Chawathe, S.; Garcia-Molina, H.; Hammer, J.; Ireland, K.; Papakonstantinou, Y.; Ullman, J.; and Widom, J. 1994. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of IPSJ Conference*. Tokyo, Japan. pp. 7–18.

[4] Church, K. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of ACL-89*. pp. 76–83. Vancouver, Canada.

[5] Doan, A.; Domingos, P.; and Halevy, A.Y. 2001. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proceedings of SIGMOD-2001*. pp. 509–520. Santa Barbara, CA.

[6] Hovy, E.H. 2003. Using an Ontology to Simplify Data Access. In *Communications of the ACM,* Special Issue on Digital Government. January.

[7] Kang, J. and Naughton, J.F. 2003. On schema matching with opaque column names and data values. In *Proceedings of SIGMOD-2003*. San Diego, CA.

[8] Levy, A.Y. 1998. The Information Manifold approach to data integration. *IEEE Intelligent Systems* (September/October), 11–16.

[9] Pantel, P. and Lin, D. 2002. Discovering word senses from text. In *Proceedings of SIGKDD-02*. pp. 613–619. Edmonton, Canada.

[10] Salton, G. and McGill, M.J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.

[11] Shaw Jr., W. M.; Burgin, R.; and Howell, P. 1997. Performance standards and evaluations in IR test collections: Cluster-based retrieval methods. *Information Processing and Management*, 33:1–14.

[12] Tova, M. and Zohar, S. 1998. Using schema matching to simplify heterogeneous data translation. In *Proceeding of VLDB-1998*. pp. 122–133.