An Information Theoretic Model for Database Alignment

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 that government staff cannot effectively locate, share, or compare data across sources, let alone achieve computational data interoperability. 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 column correspondences across environmental databases. We have seen particular success in an information theoretic model, which we call SIfT, which performs data-driven column alignments. We have applied SIfT to mapping Santa Barbara and Ventura County Air Pollution Control Districts' 2001 and 2002 emissions inventory databases with the California Air Resources Board statewide inventory database. The application of SIfT yielded 75% precision and 72.2% recall on the column alignment task. On a task of integrating new district data with the statewide database, we achieved 55% accuracy for Ventura County and 59% accuracy for Santa Barbara County.

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. 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.

To date, most approaches to wrap data collections, or even to create mappings across comparable datasets, require manual effort. Despite some promising recent 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.

Viewing the data mapping problem as a variant of the cross-language mapping problem in Machine Translation, we employed 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 identify the most informative data elements and then match data sources that share these informative elements. For example, we expect that the word "*the*" will be present in many different columns. However, consider some word, like "*carbon*", which occurs in very few columns. A random pair of columns from two data sources that both contain the data element "*the*" are intuitively not as similar as if both columns contained the data element "*carbon*". Our model automatically detects that "*the*" is less informative than "*carbon*" and will consequently assign a higher similarity to two columns that share only "*carbon*" rather than only "*the*".

This work has the potential to significantly reduce the amount of human work involved in creating single-point access to multiple heterogeneous databases.

The remainder of this paper is organized as follows. In the next section, we review related work in database alignment. Section 3 describes the environmental databases that we use as a testbed for alignment and in Section 4 we present our information-theoretic model for alignment. Our experimental results are presented in Section 4.4. Finally we conclude with a discussion and future work.

2. Related work

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% precision & recall over the past decade, determined at the annual TREC[1] conferences).

For conventional databases, 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" view method [2][3] assumes that 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" view 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 (semi-) automated methods are used to find the mappings. Schema-based matching algorithms [13] 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 methods often fail. Instance-based matching algorithms align databases using the actual data [5]. Such data driven 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], whose work most resembles ours, 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, but for instance-based matching. Instead of matching highly associated columns, which requires seed alignments, we find the data elements that are most highly associated to each column and then match columns that share these important data elements

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

3. Environmental databases

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 annually integrate 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 for use in tracking the effects of national air quality policies.

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 testbed for this research consists of the 2001 and 2002 Santa Barbara County Air Pollution Control District (SBCAPCD) and Ventura County Air Pollution Control District (VCAPCD) emissions inventories, two of the 35 California air districts.

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 automatically detects 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 measured in *SIfT* using an information theoretic model 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 the 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 the 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. We implemented a total of 20 feature domains.

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 (using KL-divergence) 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 (column) *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:

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

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 [9]:

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

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 [11]:

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

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-valued features. Hence, given a column containing all California EPA facilities and another containing only Santa Barbara facilities, cosine will find a similarity even though all non-Santa-Barbara facilities will have frequency 0 in

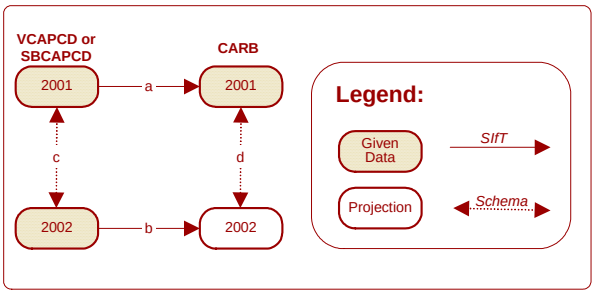

Figure 1. Experimental design for automatically generating a data transfer between VCAPCD/SBCAPCD and CARB for 2002 given historical 2001 data.

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

Applying the clustering algorithm described in [9], *SIfT* generates a similarity matrix containing the cosine similarity between each pair of columns across databases. For each column from source database *A*, we simply align it with its most similar columns from target database *S* such that the similarity between the pair of columns is above a certain threshold θ.

4.4. Scalability

Computing the similarity matrix described in Section 4.2 is daunting for large element and feature spaces. The complexity of a brute force algorithm is $O(n^2f)$, where n is the number of elements and f is the feature space.

However, a simple heuristic drawing on the properties of mutual information drastically reduces the actual running time of the algorithm. For each element *e*, we must compute its similarity with every other element by comparing their feature vectors. By sorting the features of element *e* in decreasing order of mutual information and applying a conservative minimum threshold (e.g., we used a threshold of 0.5 in our experiments), we can reduce the feature vector to only the most informative features. Using reverse indexing on the features, we need only compare *e* with the elements, *E*, which share at least one of *e*'s features. Remember that a feature of an element will have high mutual information only if that feature does not coexist with many other elements. Consequently, the size of *E* will be much smaller than the total number of elements *n*.

We use the above heuristic in our system. Another option is to use randomized algorithms, which have been shown to reduce the complexity of cosine from $O(n^2f)$ to $O(nf)$; see [10] for details.

5. Evaluation

Given two heterogeneous databases, the goal of our task is to automatically generate the same alignment that a human expert would generate. A step forward is to greatly reduce the number of alignment decisions considered by a human expert.

We evaluate our system on environmental databases. In the next section, we describe our experimental setup. In Section 5.2, we measure the precision and recall of *SIfT* alignments against a manually constructed gold standard as well as measure the reduction in human effort to generate a manual alignment. Then, in Section 5.3, we measure *SIfT*'s accuracy in automatically integrating 2002 California air quality districts' data with the California-wide emissions inventory database using historical data.

5.1. Experimental setup

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) and Ventura County Air Pollution Control District (VCAPCD). Both 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 inventories for 2001 and 2002.

To be of practical use to our governmental partners, our challenge lies both in the post-analysis of a data transfer between district and state and on integrating 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 integration.

Figure 1 shows our experimental design for automatically generating a data transfer between California districts and CARB for 2002 given historical 2001 data. First, applying *SIfT* as a post-analysis to the 2001 transfer (arrow *a*), we learn the mappings between the data columns for 2001 (evaluation of this step is shown in Section 5.2). Then, given the schema

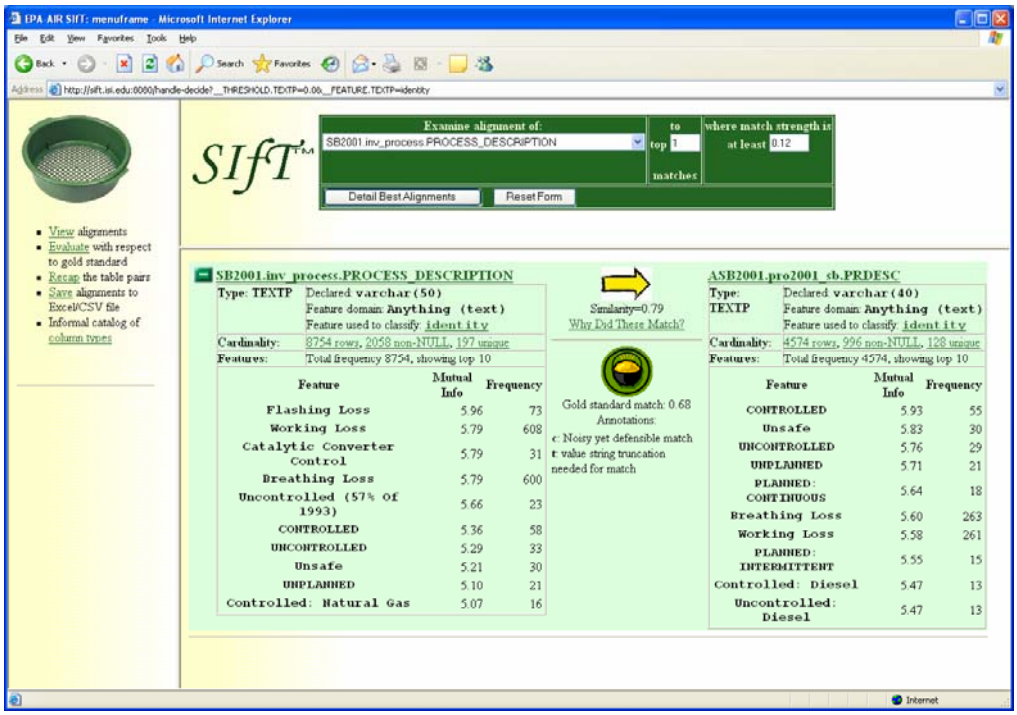

Figure 2. A correct alignment discovered by *SIfT* between the *Process Description* columns in the SBCAPCD and CARB databases. Here, the feature type is "TEXTP" so the fully qualified features are "TEXTP:Flashing Loss", "TEXTP:Working Loss", …

Figure 3. A view of the feature vectors for the *Process Description* columns in the SBCAPCD and CARB databases. The features are sorted in descending order of mutual information. Underlined features are shared by both columns whereas dark gray features are solely from the SBCAPCD column and light gray features are solely from the CARB column. *SIfT* aligns the two columns because they share many high-mutual information features.

changes for both the districts and CARB for 2002 (arrows *c* and *d*), we determine which mappings from *a* still hold. It is not unreasonable to expect that arrows *c* and *d* can be obtained from the district and CARB since schema changes are usually tracked from year to year within a department. Given a district's 2002 data, we can then follow the arrows *c*, *a*, and *d*, to integrate it with the CARB 2002 database, which is arrow *b* (evaluation of this step is presented in Section 5.3).

5.2. Alignment results

In this section, we evaluate *SIfT*'s ability to align columns across data sources (arrow *a* in Figure 1). For the purposes of this evaluation, we focused on the 2001 SBCAPCD and 2001 CARB data. The SBCAPCD and CARB emissions inventory databases used in our experiments each contain approximately 300 columns, thus a completely naïve human must consider approximately 90,000 alignment decisions in the worst case.

5.2.1. Gold standard

We manually aligned the SBCAPCD and CARB emissions inventories from year 2001. This alignment noted and estimated the strength of probable intra-source matches (e.g., likely foreign key or other join relationships) as well as inter-source links (typically, equivalence or subset relationships between columns of the two different databases). While some table-table correspondences and row-row partial equivalences were detected, the primary recorded results consisted of inter-source column associations.

The methodology for constructing the "gold standard" alignment was informal. 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 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/.

5.2.2. Precision and recall

SIfT outputs for each column in a source database the columns to which it aligns in the target database[2]. Figure 2 illustrates a screenshot of a correct alignment discovered by *SIfT* between the *Process Description* columns in SBCAPCD and CARB. Figure 3 illustrates why these columns are aligned by *SIfT*. It shows a view of the feature vectors for both columns. The features are sorted in descending order of mutual information; underlined features are shared by both columns whereas dark gray features are solely from the SBCAPCD column and light gray features are solely from the CARB column. *SIfT* aligns the two columns because they share many high-mutual information features.

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, which 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 [12], 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 data elements it contains. The *Trigram* representation extracts letter trigram features for each field whereas the *Rich* representation extracts all

[2] A 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.

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%

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.

Curiously, the *F*-measure of the simple representation is higher than the more complicated representations. 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. By inspection, we see that the feature domains in the *Rich* representation are only useful if they have very high precision and recall (e.g., zip codes).

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. Considering only two candidate alignments for each possible column will greatly reduce the number of possible decisions made by a human expert. Assuming that each of the 90,000 candidate alignments 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.

5.3. Projecting *SIfT* alignments

In the previous section, we evaluated *SIfT*'s ability to align columns across databases. Now, we evaluate the task of integrating new data as it becomes available each year (arrow *b* in Figure 1). Using the design in Figure 1, we automatically integrated 2002 VCAPCD and 2002 SBCAPCD databases with CARB's 2002 database using historical 2001 data. Unlike in Section 5.2, since CARB provided us with their 2002 databases, we have a true gold standard against which to compare our integration.

For both VCAPCD and SBCAPCD, we randomly sampled 50 columns in the automatically integrated CARB 2002 databases. A human judge was asked to classify each aligned column according to the following guidelines:

Correct: The column is aligned correctly according to the gold standard.

Partially Correct: The aligned column is a subset or superset of the gold standard alignment. This situation arises when only a selection of the column is transferred to CARB or when a join must be performed on the district tables to match the CARB schema. We must look beyond simple column alignments to solve these problems, which is beyond the scope of this paper.

Incorrect: The column is not aligned correctly according to the gold standard.

Table 3 shows the results of our evaluation. The accuracy of the system is computed by adding one point for each *correct* alignment, half a point for each *partially correct* alignment, no points for each *incorrect* alignment, and then dividing by the sample size.

Some district columns do not get integrated into the CARB database (i.e., *SIfT* does not find any alignment for these columns). In our 50 random samples for

Table 3. Evaluation results for automatically generating a CARB 2002 database from VCAPCD and SBCAPCD 2002 databases. A human judge evaluated random column alignments against a gold standard provided by CARB.

	Sample Size	*Correct*	*Partially Correct*	*Incorrect*	***Accuracy****
VCAPCD	50	25	5	20	**55%**
SBCAPCD	50	22	15	13	**59%**

* Alignments judged as partially correct count ½ points towards the accuracy.

Table 4. Accuracy of the *Top-K* alignments, according to the similarity metric described in Section 4.2, for the 50 random samples from VCAPCD and SBCAPCD.

	Top-1	*Top-5*	*Top-10*	*Top-25*	*Top-50*
VCAPCD	100%	100%	60%	70%	55%
SBCAPCD	100%	100%	95%	76%	59%

VCAPCD, nine columns were left unaligned by *SIfT*, of which six were correct and three were incorrect.

Error analysis shows that *SIfT* is particularly bad at aligning binary (Yes/No or 0/1) columns. Here, the mutual information vector-space model is not useful since binary values are shared by many columns. Such columns, which are easily identified, should be aligned by a separate process. For example, we might simply compare the ratio of 0's vs. 1's or even compare the raw frequency of 0's and 1's. Likely, however, more complex table and row analysis is needed. A possible avenue for future work is to use Kang and Naughton's algorithm [7], described in Section 2, to align these uninformative columns using the other alignments discovered by *SIfT* as seeds.

Each *SIfT* alignment includes a similarity score, as described in Section 4.2. This similarity can be viewed as *SIfT*'s confidence in each alignment. For both VCAPCD and SBCAPCD, we sorted the 50 randomly sampled alignments in descending order of *SIfT* confidence and measured the accuracy for the Top-*K* alignments, for $K = \{1, 5, 10, 25, 50\}$. Note that for binary columns, *SIfT* disregards the similarity score and assigns a 0 confidence score. The results are illustrated in Table 4. As expected, the higher the confidence *SIfT* has in a particular alignment, the higher the chances that this alignment is correct.

6. Conclusions and future work

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 and Ventura County Air Pollution Control District's 2001 and 2002 emissions inventory databases with the California Air Resources Board statewide inventory database. *SIfT* yielded 75% precision and 72.2% recall on the column alignment task. On the task of integrating new district data with the statewide database, we achieved 55% accuracy for Ventura County and 59% accuracy for Santa Barbara County.

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 can 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.

7. 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] Ravichandran, D.; Pantel, P.; and Hovy, E. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Functions for High Speed Noun Clustering. To appear in *Proceedings of Association for Computational Linguistics (ACL-05)*. Ann Arbor, MI.

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

[12] 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.

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