# Unstructured information integration through data-driven similarity discovery

**Rema Ananthanarayanan**
IBM Research, India
arema@in.ibm.com

**Sreeram Balakrishnan**
IBM Software Group, US
sreevb@us.ibm.com

**Berthold Reinwald**
IBM Research, Almaden, US
reinwald@almaden.ibm.com

**Yuen Yee**
Nuance Communications, Inc
yuenyee.lo@nuance.com

## Abstract

Information integration from multiple heterogeneous sources is one of the major challenges facing enterprises and service providers today, and one of the important problems in this domain is the integration of structured and unstructured (or text) data. In this paper we describe our work on a data-driven approach to integrating various sources of text data, without relying on the availability of schema information. To this end, we have used various existing tools from natural language processing, data mining and related areas in a novel manner. The tools are used at the 'preprocessing' stage to (a) characterise each set of unstructured information (or collection of text data), (b) identify the related sets of unstructured information and (c) relate these sets to various reference data sets. All these steps are based solely on the instance values of the data sets. Subsequently the information compiled in the preprocessing stage may be used at query time to query the structured and text data. We also present our results on applying our techniques for data integration across multiple unstructured data sources, relating to customer comments of a service provider.

## 1 Introduction

Most techniques developed today for data integration across heterogeneous data sources operate on structured or categorical data, usually made available in relational databases. However a huge proportion of business data resides in unstructured documents spread across the enterprise, such as emails, spreadsheets, facsimiles and other sources.[1]. Non-conventional sources such as blogs and third-party review sites are also increasingly serving as rich sources of information on trends and opinions for business intelligence. One of the key challenges that enterprises face today is being able to automatically integrate the information from these various heterogenous sources, and query this information seamlessly across the structured and text data, for extracting business intelligence. In our work here, we look at the problem of integrating various sources of unstructured information, using only data-driven techniques. Current approaches to data integration based on instance values operate at entity level or record level and we extend the approaches to data set linking. Our focus is on being able to compare multiple sets of text data items, analogous to comparing across columns in database tables. Further, just as each column element in the database may be characterised by an 'attribute' (which could typically be the column name), each data set may also be characterised by one or more attributes. Multiple data sets across different data sources, or even within the same data source, may possess the same attributes. Further, each data set also has 'value' elements that correspond to the individual instance values comprising that set. When querying for an attribute across data sets from different sources, it is therefore necessary to ensure that all the data sets described by that attribute are included. In our work, we achieve this by clustering the related data sets based on the data contents rather than the column or data set names. Our motivation is

1. Purely data-driven approaches appear more amenable for complete end-to-end automation and
2. Where feasible, these methods may subsequently be supplemented with schema-based integration techniques to achieve better results.

Different techniques exist to measure the degree of similarity between two or more data sets based on the metadata (or schema information) in most cases, and in a few cases, based on the actual data values themselves. However, these techniques have mainly been restricted to structured or categorical data. Our overall goal here is to *Provide a means for data-driven similarity discovery across multiple sources of unstructured data, so that the discovered information may be integrated with the existing schema of the structured information, allowing querying across the structured and text data.*
Our solution comprises the following steps:

- Identify groups of related data sets based on various text-processing and data mining techniques;
- Identify attributes of the related data sets, based on comparison with domain-specific reference sets and keyword generation;
- Present a view of the various data sets as a single repository, for subsequent querying.

[1] Some studies estimate that more than 80% of the data in enterprises is unstructured data

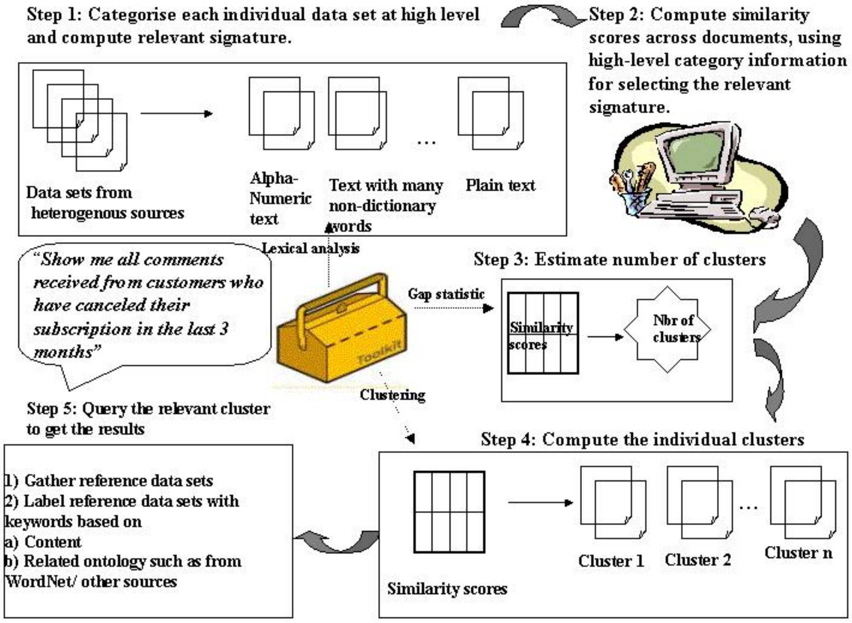
Step 1: Categorise each individual data set at high level and compute relevant signature.
Step 2: Compute similarity scores across documents, using high-level category information for selecting the relevant signature.
Data sets from heterogenous sources
Alpha-Numeric text
Text with many non-dictionary words
Plain text
...
Lexical analysis
"Show me all comments received from customers who have canceled their subscription in the last 3 months"
Toolkit
Step 3: Estimate number of clusters
Gap statistic
Similarity scores
Nbr of clusters
Clustering
Step 5: Query the relevant cluster to get the results
1) Gather reference data sets
2) Label reference data sets with keywords based on
a) Content
b) Related ontology such as from WordNet/ other sources
Step 4: Compute the individual clusters
Similarity scores
Cluster 1
Cluster 2
Cluster n

Figure 1: Flow of tasks in the pre-processing stage, to build an integrated view

The various preprocessing steps (described in detail subsequently) include top-level identification of the type of data set, signature computation of each data set, similarity computation across pairs of data sets of the same top-level category, clustering of related data sets based on the similarity measures and making available the discovered information in a structured format, for subsequent querying.

Figure 1 gives a system overview of the end-to-end solution and the details are discussed in section 3. In Section 2 we describe the earlier work on integration of text data and where our approach differs. In Section 3 we describe the various steps in our system in detail. In Section 4 we discuss the application of our techniques to a real-life business scenario. We conclude in Section 5 with our observations and future work.

## 2 Background

Semantic integration of structured and unstructured data is an active area of research for its potential impact on solving a wide range of real-life problems. Traditional approaches to data integration have looked at mapping various schemas to a global schema or matching the schemas to match different data sets [5][14]. Other related works include [11], [7], [2], [3], [13]. In the absence of schema information, instance value matching is needed to relate or link data sets, as in [4] which describes graph-based data-matching techniques for semantic integration of structured data. In [10] the authors describe their work on matching textual attributes using text similarity. In [16] the authors discuss entity retrieval over text and structured data, to locate data fragements that talk about the same entity. Similarity matching is also used in [9] to create meaningful associations of information on a person's desktop and provide improved search. Similarly, in [12] a data-driven approach is used based on an information-theoretic model, to match data sources. In [8] the system described uses signatures and summaries to mine the database structure and determine fields with similar values. Most of these works study issues in integrating structured data sets, while integration of text data is in the context of existing structured or relational data, as also in the case of [6] and [15]. In our work here, we extend the notion of signatured described in [4] to unstructured data and describe techniques for constructing signatures for text data sets, and comparing them subsequently using similarity measures. This allows us to subsequently link together various data sets for which we do not earlier have any metadata. In the next section we describe our end-to-end system for linking unstructured data sets.

## 3 System overview

Figure 1 shows the various pre-processing steps required for the integration of the various data sets, to facilitate subsequent querying across the structured and unstructured sets. We borrow tools and techniques from text-processing and data mining as relevant, in the various pre-processing steps. While none of the individual steps in themselves are new, we are not aware of any previous application that combines these tools in this novel manner to achieve an end-to-end instance-value based integration.

### 3.1 Lexical analysis and signature generation

A signature summarises a specific characteristic of a data set that is pertinent to evaluating the similarity with other data sets. Signature computation is an important step in the

*character bigram*: (th,hi,is,te,es,st,se,en,nt,nc,ce)
*character trigram*: (thi,his,tes,est,sen,ent,nte,ten,enc,nce)
*word unigram*: (this,is,a,test,sentence)
*word bigram*: (this is,is a,a test,test sentence)

Table 1: Sample signatures for *'This is a test sentence.'*

data-driven integration of structured data sets [4]. The notion of signatures may be extended to unstructured data sets; signatures may be defined at different levels of granularity, for instance, word, character and morpheme, and in different sizes such as unigram and bigram. Table 1 shows various signatures for a sample sentence. At the top-level, heterogenous data sets could comprise many types of data, including numeric, plain text, alpha-numeric text, plain text with a large or small percentage of alpha-numeric text and other data types. Each of these types is amenable to a different signature type, for comparison with other data types. Hence, as a first step, we identify a top-level category for each data type, as shown in Figure 2, in order to understand the more relevant signatures to generate. We then compute the signatures of the individual data sets and use these signatures to compute similarity values across different sets of data. We use a lexical analysis tool, LanguageWare, which is a natural language processing solution developed in IBM, for tokenising and subsequent preprocessing. (http://www-306.ibm.com/software/globalization/topics/languageware/index.jsp). Different routines were also developed for character-level bigram and trigram signatures. At the end of this step, we have associated with each data set, a top-level category to which it belongs and the relevant signature.

### 3.2 Relating data sets based on similarity computations

The $cosine$ similarity with TF-IDF (Term Frequency-Inverse Document Frequency) weighting is widely used for document similarity assessment in search, question-answering and information retrieval applications [1]. We extend this to data sets, where each data set is represented in one of the many possible signature representations, as determined by the top-level category of the data set. The similarity scores of pairwise data sets is calculated across each pair in each of the top-level categories. We initially computed similarity scores based on the different signatures on some test data sets that we manually assessed to be similar. Our observations were:

- Similarity values based on character trigram signatures give good results for data sets that contain many names and addresses.
- Similarity values based on character bigram signatures for general text data are high, but this is not discriminatory and also shows many false positives.
- Similarity values based on word unigram signatures seem to be representative for plain text documents, when compared to manual observations. Word bigrams appear representative in some cases, but appear too discriminatory and present false negatives in some cases.
- Similarity values for product data sets do not appear to be representative of the actually observed sets, for the different signatures. This could be because the product data sets that we have been looking at have a high proportion of non-dictionary words, alphanumeric strings, numbers and abbreviations. This would require similarity measures to be based on other definitions, for example pattern-based or regular-expression based. These special signatures are explored elsewhere.

At the end of this step, each data set is associated with $n_c$ similarity values, where $n_c$ is the number of data sets in the top-level category that this data set belongs to.

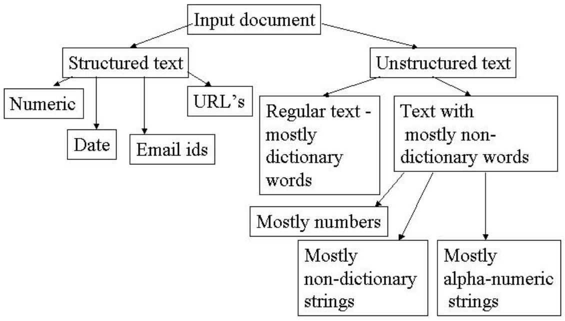
Input document
Structured text
Unstructured text
Numeric
URL's
Date
Email ids
Regular text - mostly dictionary words
Text with mostly non-dictionary words
Mostly numbers
Mostly non-dictionary strings
Mostly alpha-numeric strings

Figure 2: Toplevel categories of data sets

### 3.3 Identifying the related data sets

Given the large number of unstructured data sets, we need to group them into related categories or clusters for efficient querying of related sets. Our motivation for clustering the data sets rather than viewing them as isolated data sets is that typically data in enterprises fits into one or more of a finite number of domain-related topics. Further, with multi-channel business operations, it is increasingly common to have similar data sets from multiple sources. For instance (as we will also see in the sample application subsequently) an enterprise could receive customer feedback over phone, by email, at point-of-sale and over the web. Clustering the various data sets is an effective means of grouping related data for subsequent querying, in the context of specific enterprise operations. Since each cluster is identified with one or more keywords or attributes (described in section 3.4), the query attributes are mapped to the clusters, and the individual data sets in each cluster are then queried to fetch the results.

**Estimating the number of clusters**

Clustering tools require as input either the number of clusters or a threshold value, for each cluster. Since we wish to reduce the number of manual inputs, we use the gap statistic technique [17] to estimate the number of clusters in the heterogeneous data sets, and we then use this estimate as input to the clustering tool, to compute the actual clusters. Gap statistic uses a statistical procedure to estimate the optimal number of clusters by comparing the change in within cluster dispersion to that expected under an appropriate reference null distribution. The input is the set of similarity measures for the data sets under consideration. The algorithm is invoked to cluster the data sets, whenever one or more data sets are added to the system. The similarity values obtained in Section 3.2 are input to the tool, and the output is the optimal number of clusters.

**Clustering the data sets**

We use the the number of clusters computed in the previous step as inputs to the clustering tool *CLUTO* [18], along with the various pairwise similarity values of the data sets, to cluster the various data sets. At the end of this step, each data set is associated with a specific cluster.

### 3.4 Matching with reference data sets

In instance-based integration, there is no way of knowing apriori the topic that each data set pertains to. Clustering tools such as CLUTO optionally return keywords that may be associated with each cluster, when the individual data sets are input to the tool. Cues from metadata and schema information, where available, can also be used. However, to automate the data-driven integration process, we additionally use a repository of reference data sets in order to match clusters to keywords or topics. This requires building domain-specific reference data sets. (In our instance, for the various test data sets, we have used a fraction, upto 25%, as the reference data set, for the various topics. As the number of domain-specific reference data sets increase in our repository, it would be possible to increase the level of automation for the end-to-end scenario to more domains.) Each reference data set is hand-labeled by one or more initial keywords, based on manual observation, and our knowledge of the data sets. Subsequently, the set of keywords has been expanded (a) based on keywords from the contents of the data sets, though for our example this did not seem very productive, and (b) based on an ontology built from synonyms/ related taxonomies of the initially selected label. For instance, the reference data set containing customer comments is labeled with the keywords ‘comment’ and ‘complaints.’ Subsequently, keywords like ‘feedback’ and ‘suggestion’ are added to the set of labels, based on synonyms of the initially set keywords. Some of the advantages of using reference data sets are

1. The similarity with the reference data sets helps us validate our clustering, since we do not have any pre-defined thresholds for our clustering.
2. The use of reference data sets helps us identify new sets of topics as they arise in the corpora, since the new data sets would not be close to the reference data sets in similarity value.
3. Extensive sets of reference data would help increase the degree of automation as new domains are handled.

The clustered data sets are now compared with each reference data set, in order to identify and relate the clustered sets with some topic or keyword. At the end of this step, we are able to associate each data set with a set of keywords or labels relating to the cluster to which the data set belongs. We now have a system where we can query on the attributes of each unstructured or text data set.

### 3.5 Schema

The list of top-level categories supported, the relevant signature types for each top-level category, the reference data sets and the key words associated with each reference data set are inputs made available at pre-processing. Information computed during pre-processing, such as the top-level category for each input data set, the signature for each data set, pairwise similarity value for each pair of data sets of the same type, association of each data set with a cluster and association of a cluster with a reference data set are other pieces of information that are generated and used during subsequent querying across the data sets.

## 4 An application

| |
|---|
| **UST5** |
| Wrong commitments on X by salesperson |
| Bill not recd on time and sales person makes wrong commitment |
| Says the service is good |
| **UST7** |
| Sub wants to know about scheme X |
| Subs is getting error XXX |
| Customer wanted to have details regarding scheme Y; agent took the details and updated the cust. rgding the same. |
| **UST8** |
| Able to communicate clearly and professionally |
| Clarity and communication can be better |
| **UST10** |
| Customer issues resolved online |
| Partial issues resolved online |
| Customer issues not resolved online |
| **UST11** |
| Spoke to X today; as per him, he has not received the welcome kit till today; plz check and do whatever is necessary. |
| Customer is not satisfied with the explanation given by the sales executive; induction visit also has not happended; |
| Customer suggests to book an appointment before visiting |

Table 2: Sample text data

We describe here the application of our techniques to solve a problem in the customer-relationship domain. Typically, information about customers is available from multiple sources. Different organizations within the enterprise may be managing the information at the different stages (in some cases, some of the activities like the post-sales support could be outsourced to a third party) and the information is therefore available in different formats and on different host applications. One example is, enterprises often need to access all information related to a customer in one view, but this is not always available in a straightforward manner. We give below a sample scenario and subsequently present the results of our approach for this problem. While we have tested our methods on a small data set that was available to us, all our methods are readily extendible to larger sets and to non-traditional sources such as blogs and online reviews.

### 4.1 Sample Scenario

A service-provider company receives records of service calls from multiple sources, such as web complaints, emails, telephonic complaints and service calls from possibly multiple outsourced call centers. The end goal is to provide a unified view of all the information pertaining to each customer so that queries may be run on this unified view. In one instance, eight different sources of information were gathered for consolidation. The information represented inputs gathered at different points in time - when the customer was registering for service, when the customer called up the call center for assistance, or when the questionnaires were sent proactively by the service provider, seeking feedback.

Given this scenario, we want to have an integrated view of the information that can enable execution of queries such as *'Show me all complaints from customer X from all sources'*. This would require an integrated view of all the text data sets, to identify the multiple sources representing complaints.

## 4.2 Data sets

We use the terms $UST1, UST2, \ldots$ to refer to the various sets of UnStructured Text in the data sources. Specifically, there were eight sources of information, each containing multiple structured and text data sets. Our focus was on the text data sets, which include complaint classification, follow-up action classification, complaints from customers, feedback on service, transcripts of customer calls, resolution of issues and other information. Here we have not included text data such as addresses and customer names. The text data sets were spread across the various data sources as under:
Source 1: UST1,UST2,UST3,UST4,UST5
Source 2: UST6
Source 3: UST7, UST8, UST9, UST10, UST11
Source 4: UST12
Source 5: UST13
Source 6: UST14
Source 7: UST15, UST16, UST17
Source 8: UST18
Table 2 shows some sample data from the various customer data sets. This data has been presented in a suitable anonymized form for reasons of high confidentiality.

Apart from these eight sources, we also use other data sets for testing our assumptions and validating each step of the processing. This was required since in practice data sets would be much larger and possibly more heterogeneous than what we have shown here. Further, we wished to validate the use of tools such as lexical analysis, clustering and gap statistic, in the context of information integration and preprocessing for querying the integrated information. The additional data sets would also serve to play the role of noise in real-life systems.

## 4.3 Additional data sets

The complete data sets used include:

1. Real-life data from a service provider, described in detail in section 4.2, with some samples shown in table 2.
2. Product data from the IT department of a large enterprise (this was a very large set and we have used only a subset for our experiments.) This again comprised various data sets such as product ids, product names, brief and long descriptions of products, and other related data sets.
3. Movie reviews downloaded from the web
4. A collection of presidential inaugural and union address speeches available from Natural Language Toolkit (http://nltk.sourceforge.net)
5. A collection of financial documents - annual and quarterly filings of some companies.

| | Reference dataset | | | |
|---|---|---|---|---|
| **Cluster** | **Finance reports** | **Movie reviews** | **Speeches** | **Customer Info** |
| **Finance reports** | **0.544** | 0.088 | 0.205 | 0.183 |
| **Movie reviews** | 0.085 | **0.267** | 0.189 | 0.172 |
| **Speeches** | 0.189 | 0.221 | **0.615** | 0.302 |
| **Customer info** | 0.185 | 0.188 | 0.313 | **0.637** |

Table 3: Similarity scores between the reference datasets and automatic clustered data.

The sizes of the individual data sets varied widely, varying from a few 100 entries to many thousands of entries.

## 4.4 Results and observation

Each data set was processed to determine the top-level category as shown in Figure 2. Word-level signatures were used for the unstructured data sets with plain text, and character trigram signatures for text with large proportion of non-dictionary and alphanumeric strings.

From the high-level classification, data from the *financial reports*, *movie reviews*, *speeches* and *some subsets from the service provider data sources* were classified as plain text. We generated the word unigram signatures and computed the pair-wise similarity scores for each data set in this category. We then used the gap statistic tool to estimate the optimal number of clusters in each of the toplevel categories. The optimal number of clusters for the plain text data sets was computed as four, by the gap statistic tool. Subsequently, these data sets were run through the clustering tool *CLUTO*, with the input parameter of 4 clusters. The results in terms on the number of clusters and the actual cluster contents appear reasonable. While this is to be expected in terms of the non-overlapping nature of the data sets that we used, we are seeking to get more real-life data to further validate our technique.

In section 3, we describe techniques for labeling the clusters using reference data sets. Table 3 shows the similarity scores of the reference dataset and the automatic clustered data. We see that for the clusters labeled financial reports, speeches and customer feedback of service providers, the similarity scores are high (greater than 0.5) when the reference dataset and the cluster are of the same topic, while the similarity score of the off-topic cluster is low (less than 0.2). Hence we are able to discriminate between the different clusters with confidence. For the movie reviews, the similarity score is comparative low; one explanation is that the individual reviews are very different, varying based on genre and the cast. Therefore, using the initial 20% of reviews as a reference is probably not adequate to represent the cluster. If the similarity score between any reference dataset and a particular cluster is low, we can identify this cluster as a new cluster, and then assign keywords of this cluster.

Based on the clustering results and the mapping with reference data sets, data sets $UST1$, $UST2$, $UST4$,$UST13$, $UST14$ and $UST17$ were identified with the keyword 'com-

| Customer | Dataset | Feedback |
|---|---|---|
| **Id1** | UST17 | No issues |
| | UST14 | Officer has time till tomorrow morning |
| | UST1 | Installation engr done work neatly |
| | UST4 | Customer says good service |
| | UST2 | DSL speed line very slow and every half hour disconnects |
| **Id2** | UST17 | Pls honor when committed |
| | UST14 | Problem X was reported, was committed 4 times that one of customer care execs will turn up to his place but no one had turned up |
| | UST1 | Overall it is good |
| | UST4 | Now no problem |
| **Id3** | UST13 | Spoken to Mr X - as per him, satisfied with services |
| | UST1 | Good |

Table 4: Sample output for the query: Return all feedback received from each customer from all sources OR with NEWT1(id, comments1,comments2, comments3) as (select id,UST1,UST2,UST4 from T1) select T7.id,T7.UST17,table1.comments1, table1.comments2,table1.comments3 from NEWT1,T7, where NEWT1.id = T7.id

ments'. In this example, using the customer id from the structured data as index into the customer, we obtained the query results as shown in Table 4. We have imported the various sources of information into relational tables, and run the query across these tables. While the sample data sets divide into neat clusters with little overlap, lack of additional real-life data prevented us from testing our approach on larger sets.

## 5 Conclusions

In this paper we discuss a data-driven approach for data integration across multiple sources of data. Our approach is driven by the actual instance values of the data, without relying on any metadata information. Further, we have attempted to link the generated schema from the unstructured data sources with schema from the structured data sources, where available. In putting together our end-to-end solution, we have used techniques already available, such as gap statistic and clustering, along with development of techniques for signature and similarity computation, high-level categorization, and steps for labeling of clustered data sets. We are not aware of any other novel application of these existing techniques to solve an enterprise problem of high import. We have presented our results on some sample data made available to us. Though this is a small set, the results are promising and appear scaleable for use on larger data sets.

We plan to validate our techniques on a larger and a wider variety of data repositories. We also plan to look further into different types of signature generation for different types of unstructured data sets, such as pattern-based signatures in the case of text with a large percentage of alphanumeric words.

## References

[1] R. Baeza-Yates and B. Ribeiro-Neto. Modern information retrieval. In *ACM press*, 1999.

[2] P. Bernstein. Applying model management to classical meta data problems. In *CIDR*, 2003.

[3] P. Bernstein, S. Melnik, C. Quix, and M. Petropoulos. Industrial-strength schema matching. In *SIGMOD Record*, volume 33, 2004.

[4] P. Brown, P. Haas, J. Myllymaki, H. Pirahesh, B. Reinwald, and Y. Sismanis. Toward automated large scale information integration and discovery. In *Modern Issues in Data Management*. Springer Verlag, 2005.

[5] M. L. C. Batini and S. Navathe. A comparative analysis of methodologies for database schema migration. In *ACM Computing Surveys 18*, 1986.

[6] V. Chakravarthy, H. Gupta, P. Roy, and M. Mohania. Efficiently linking text documents with relevant structured information. In *VLDB*, 2006.

[7] W. W. Cohen. Data integration using similarity joins and a word-based information represeintation language. *ACM Transactions on Information Systems*, 18(3):288–321, July 2000.

[8] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *SIGMOD*, 2002.

[9] X. Dong and A. Halevy. A platform for personal information management and integration. In *CIDR*, 2005.

[10] A. Koeller and V. Keelara. Approximate matching of textual domain attributes for information source integration. In *Proceedings of the Second International Workshop on information quality in information systems*, pages 77–86. ACM SIGMOD, 2005.

[11] J. Madhavan, P. Bernstein, K. Chen, A. Havely, and P. Shenoy. Corpus-based schema matching. In *International Conference on Data Engineering*, 2005.

[12] P. Pantel, A. Philpot, and E.Hovy. Aligning database columns using mutual information. In *Proceedings of the 2005 National Conference on Digital Government Research*, 2005.

[13] R. Pottinger and P. A. Bernstein. Schema merging and mapping creation for relational sources. In *Proceedings of the 11'th International Conference on Extending Database Technologies: Advances in database technology*. ACM, 2008.

[14] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.

[15] P. Roy, M. Mohania, B. Bamba, and S. Raman. Towards automatic association of relevant unstructured content with structured query results. In *CIKM*, 2005.

[16] M. Sayyadian, A. Shakery, A. Doan, and C. Zhai. Toward entity retrieval over structured and text data. In *Proceedings of WIRD'04 - the first workshop on the integration of Information Retrieval and databases*, 2004.

[17] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. In *Technical Report 208, Department of Statistics, Stanford University*, 2000.

[18] Y. Zhao and G. Karypis. Hierarchical clustering algorithms for document datasets. In *Data Mining and Knowledge Discovery*, volume 10, pages 141–168, 2005.