Finding Quality in Quantity: The Challenge of Discovering Valuable Sources for Integration

Theodoros Rekatsinas
University of Maryland
thodrek@cs.umd.edu

Xin Luna Dong
Google Inc.
lunadong@google.com

Lise Getoor
UC Santa Cruz
getoor@soe.ucsc.edu

Divesh Srivastava
AT&T Labs-Research
divesh@research.att.com

ABSTRACT

Data is becoming a commodity of tremendous value for many domains. This is leading to a rapid increase in the number of data sources and public access data services, such as cloud-based data markets and data portals, that facilitate the collection, publishing and trading of data. Data sources typically exhibit wide variety and heterogeneity in the types or schemas of the data they provide, their quality, and the fees they charge for accessing their data. Users who want to build upon such publicly available data, must (i) discover sources that are relevant to their applications, (ii) identify sources that collectively satisfy the quality and budget requirements of their applications, with few effective clues about the quality of the sources, and (iii) repeatedly invest many person-hours in assessing the eventual usefulness of data sources. All three steps require investigating the content of the sources manually, integrating them and evaluating the actual benefit of the integration result for a desired application. Unfortunately, when the number of data sources is large, humans have a limited capability of reasoning about the actual quality of sources and the trade-offs between the benefits and costs of acquiring and integrating sources. In this paper we explore the problems of automatically appraising the quality of data sources and identifying the most valuable sources for diverse applications. We introduce our vision for a new *data source management system* that automatically assesses the quality of data sources based on a collection of rigorous data quality metrics and enables the automated and interactive discovery of valuable sources for user applications. We argue that the proposed system can dramatically simplify the *Discover-Appraise-Evaluate* interaction loop that many users follow today to discover sources for their applications.

1. INTRODUCTION

In the last few years, the number of data sources available for integration and analysis has increased many-fold because of the ease of publishing data on the Web, the proliferation of services that facilitate the collection and sharing of data (e.g., Google Fusion Tables [17]), and the adoption of open data access policies both in science and government. This deluge of data has enabled small and medium enterprises as well as data scientists and analysts (e.g., political or business analysts) to acquire and integrate data from multiple data sources. Although much of the data is freely available, the number of data sources that charge monetary fees for access is rapidly increasing, a trend that is expected to continue as data is further commoditized [4, 32].

Given the high number of available data sources and the fact that acquiring data may involve a monetary cost, it is challenging for a user to identify sources that are truly beneficial to her application. In fact, sources may provide erroneous or stale data [11, 23], they may provide duplicate data at different prices, and may exhibit significant heterogeneity in the representation of stored data, both at the schema and the instance level [7, 23, 9]. After choosing a set of sources to use, a substantial effort must be spent in cleaning the sources, extracting structured information from them (for unstructured sources), constructing schema mappings, resolving entity references, and setting up the overall integration pipeline for continuous ingest; hence the initial selection of sources becomes even more important. The above give rise to the natural question of how can one discover *valuable sources*, i.e., sources that maximize the user's benefit at the minimum cost. Recent work [12, 29] showed how, given a fixed data domain, the benefit of integration can be quantified using rigorous data quality metrics, such as *coverage*, *accuracy* and *freshness*, and introduced the paradigm of *source selection* to reason about the benefits and costs of acquiring and integrating data from static and dynamic sources. This line of work showed how one can identify the set of sources that can maximize the marginal gain for a predefined benefit function using a fixed quality metric or a fixed weighting scheme across different quality metrics. However, the proposed techniques are not sufficient for general users.

First, the data quality metrics used to quantify the benefit of integration are complex and it is not easy, especially for common users, to understand the trade-offs between these metrics. Having a fixed and predefined weighting mechanism among different quality metrics does not allow the user to understand the implications of the quality trade-offs for source selection and identify the set of sources that truly fits her requirements. We illustrate this using the following example inspired by recent work by Schutte et al. [30].

EXAMPLE 1. *Consider a political scientist who wants to find data providing supporting evidence for a new theory on causal relationships between interactions among different actors (including individuals, international organizations or countries) at a specific location. Such interactions are usually reported in newspapers, thus, our system should allow the political scientist to discover the newspapers whose news articles will provide her with sufficient*

7th Biennial Conference on Innovative Data Systems Research (CIDR'15) January 4-7, 2015, Asilomar, California, USA.

data either supporting or contradicting her theory. The completeness (i.e., coverage) and accuracy of data are important here but the freshness of data is not so crucial as delayed mentions of actor interactions will not affect the evidence provided by the data. While this distinction between coverage and freshness is clear, i.e., the scientist may require that freshness is completely ignored, the correct trade-off between accuracy and coverage is not well known in advance. In fact demanding only highly accurate data may limit the coverage of events significantly, thus, the user should have the flexibility to explore and understand the trade-off between accuracy and coverage.

Second, the existing source selection techniques do not allow the user to evaluate the result returned by the system. The current techniques focus on finding a single set of sources that maximizes the marginal gain between the benefit and cost of integration given a budget constraint by the user, and report the overall quality and cost characteristics to the user. This does not allow the user to understand the individual quality and cost contribution of each selected source to the final result. Providing this information and enabling the user to *interactively explore* how the source selection solution will be affected by adding or removing sources is necessary to evaluate the solutions returned by the source management system. Finally, previous techniques focus on fixed domains and do not support source selection for arbitrary applications in multi-user environments with diverse tasks.

In this paper, we introduce our vision for a *data source management system* that enables users to discover the most valuable data sources for their applications. We present how such a system can support the interactive exploration of different sets of sources, allowing the user to truly understand the quality and cost trade-off between different integration options. To enable the latter, we show how one can augment traditional knowledge bases with a *correspondence graph* to reason about the content and quality of data sources. We also extend the paradigm of source selection to a multi-objective optimization problem that allows users to make optimal decisions in the presence of trade-offs between conflicting objectives (e.g., different data quality or budget requirements).

The remainder of the paper is organized as follows. In Section 2 we provide an overview of the architecture of a data source management system, the key functionalities it should support and discuss the main challenges in building such a system. In Section 3 we present our proposed data source management system, introduce the different modules of the system and propose techniques for addressing the aforementioned challenges. Section 4 presents a demonstration outline of our prototype data source management system. Finally, Section 5 discusses related work and Section 6 concludes the paper.

2. DATA SOURCE MANAGEMENT

In this section we provide an overview of our proposed data source management system, and discuss the key functionalities such a system needs to support. Furthermore, we present the key challenges in supporting each of these functionalities.

2.1 System Overview

We envision a data source management system following the architecture shown in Figure 1. The system is composed of (i) a data extraction module, (ii) a source analysis engine and (iii) a query engine. The basic operations of a data source management system can be divided into an *offline* phase and an *online* phase. During the offline phase, the data extraction module is responsible for extracting and storing raw data from different data sources. Then, the source analysis engine analyzes this raw data to identify the content of sources, evaluates their quality with respect to a collection of data quality metrics and constructs an index describing both the content and the quality of each source. This index will be used during the online phase when a user interacts with the system and discovers the most valuable sources for her application. The aforementioned operations may be performed once, if the sources are static, or repeatedly over time, if the sources are dynamic and their content changes. During the online phase, a user gives a description of her requirements as input to the query engine which detects the most valuable sources using source selection. Typically a user would start by providing the following information: (i) a free-text description of the task corresponding to a collection of mentions to either abstract concepts (e.g., "commerce treaties") or specific instances, such as locations, organizations, people and items, (ii) a selection of relevant data quality metrics from a list of supported metrics, and (iii) a desired budget characterizing the amount of money the user can afford for acquiring data. Given these specifications, the query engine should perform the following operations:

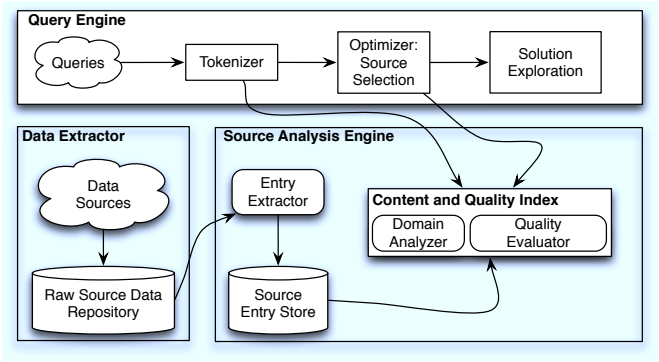

Figure 1: Data Source Management System Architecture.

1. **Discover.** The system should automatically determine which sources are relevant to the task by mapping the content of the task description to the content of sources.
2. **Appraise.** After discovering the relevant sources, the system should automatically find subsets of sources that, if integrated together, maximize the integration quality under the budget requirements of the user. If available, the system should identify multiple solutions that correspond to different trade-offs among different quality metrics.
3. **Evaluate.** The solutions discovered in the previous phase should be presented to the user together with a concise description of their quality characteristics as well as a description of the data sources included in each of them. Moreover, the user should be able to interactively explore the returned solutions either by removing sources from a recommended solution or by examining solutions with similar characteristics that contain different sources. The latter enables the user to identify the solution that is best suited for her task.

2.2 Challenges

We now discuss the main challenges in each of the operations presented above.

2.2.1 Analyzing the Content of Sources

The first challenge is analyzing the content of diverse data sources and being able to identify the data domain of each source. This is necessary for a multi-user environment where users with varying requirements interact with the system. The data source management system has to deal with a variety of diverse datasets rang-

ing from a structured Web table providing financial data to an unstructured news article extracted from a newspaper. Therefore, a data source management system should be able to reason about the semantic content of different types of data ranging from tables to DOM trees to free-text. Through that, the data source management system can enhance the source content with semantic annotations; specifically, the data provided by the source can be viewed as a collection of extracted data entries, each associated with specific entities or entity types (also called *concepts*). Data entries from sources can often be noisy (i.e., have missing or erroneous information). Moreover, data entries from different sources may correspond to the same latent data entries. Notice that only partial information is known for these latent data entries via observations provided by the sources. We refer to these latent data entries as *world data entries*.

Accessing the entire source content to do such annotations may not be possible for all sources as many require a monetary fee for accessing their data. Nevertheless, there are many cases where sources offer free samples or limited-transaction access to their content. This raises further challenges, such as how can one obtain a comprehensive view of the different concepts covered by a data source, how often should one obtain and analyze content samples to identify the rate of change for a source and how can one determine the right sample size to detect the source's focus.

2.2.2 Data Source Quality Metrics

The second challenge is determining the quality of data. Although it is possible to talk about the quality of a data source by itself, it is more natural, useful, and accurate to talk about the quality of a source with respect to some specific *context*. For example, "ESPN" is a high coverage source for "sports in the USA" but has zero coverage for politics. To formalize the notion of context we introduce the concept of a *context cluster* (c-cluster). A c-cluster defines the data domain that *corresponds* to the content of sources and is specified by as a conjunction of *a set of entities* and *a set of entity types*. For example, the entity type set and entity set corresponding to the c-cluster "Sports in the USA" are $\{Sports\}$ and $\{USA\}$, respectively. In the remainder of the paper we will refer to the set of entities and set of entity types of a c-cluster as the *domain* of the c-cluster. We point out that a c-cluster with a non-empty set of entity types can be decomposed to more specific data domain points by considering the instances of the entity types associated with it. For example, "Sports in the USA" can be further decomposed to "Baseball in the USA", "Basketball in the USA", etc. Finally, we define the content of a c-cluster as the set of world data entries associated with the domain of the c-cluster. To measure the quality of a data source with respect to a c-cluster, one needs to compare the data records provided by the source to the world data entries contained in that c-cluster. A source may provide data entries from multiple c-clusters, and thus, it can have multiple quality profiles corresponding to different c-clusters, allowing us to capture the quality of the source more faithfully, at a fine granularity.

Traditionally, the quality of data sources has been measured using the percentage of data and the amount of erroneous information provided by the source. In the last decade, however, there has been a growing interest in defining diverse metrics to assess the quality of data [27]. Nevertheless, most of these metrics are hard to quantify and measure for arbitrary datasets. Next, we focus on a collection of data quality metrics that can be expressed as probability distributions and hence admit rigorous definitions. The following list extends metrics introduced in our recent work [12, 29].

Let $\mathcal{C}$ be a context cluster, $\mathcal{C}.D$ be the domain corresponding to $\mathcal{C}$ and $\mathcal{C}.O$ be the set of all world data entries contained in $\mathcal{C}$. Notice that $\mathcal{C}.O$ is not fully known but only partial information is available for the content of this c-cluster via the data sources. We assume that each entry $e \in \mathcal{C}.O$ has a set of attribute values denoted by $e.A$. For example, these attributes may correspond to a location or an organization or a time point. Moreover, we assume that both the entries in $\mathcal{C}.O$ and their attribute values may change over time. Next, we define a collection of quality metrics for arbitrary sets of sources generalizing the case of a single source. Given a set of sources $\bar{S}$ that provide entries contained in a c-cluster $\mathcal{C}$ and an integration model F we have $F_{\mathcal{C}}(\bar{S}) \subseteq \mathcal{C}.O$, where $F_{\mathcal{C}}(\bar{S})$ denotes the set of data records that are related to $\mathcal{C}$, extracted after integrating all sources in $\bar{S}$. Using this notation we have the following metrics:

Coverage. The coverage of $\bar{S}$ with respect to $\mathcal{C}$ can be defined as the probability that a data entry e chosen at random from $\mathcal{C}.O$ will be present in $F_{\mathcal{C}}(\bar{S})$. We do not consider the correctness of attributes of the entry e when computing coverage. In case the domain of c-cluster $\mathcal{C}$ is specified by a collection of entity types, and thus, incorporates multiple points, one can extend this definition of coverage to a more generic probability distribution over the different domain points included in $\mathcal{C}$. This distribution will correspond to a multinomial distribution over all domain points where the probability value for each point is computed similarly to the overall coverage for $\mathcal{C}$.

Accuracy. The accuracy of $\bar{S}$ with respect to $\mathcal{C}$ corresponds to the probability that a data entry chosen from $F_{\mathcal{C}}(\bar{S})$ is correct with respect to $\mathcal{C}.O$. The latter means that the entry must be present in $\mathcal{C}.O$ and all its attributes mentioned in $F_{\mathcal{C}}(\bar{S})$ should have the correct values. This definition of accuracy is equivalent to the traditional definition of accuracy focusing on erroneous values [12] and the metric of freshness (i.e., the percentage of up-to-date data in a source) focusing on stale data [29]. Similarly to coverage, if the domain of $\mathcal{C}$ contains multiple points, one can extend accuracy to a probability distribution over those.

Timeliness. The timeliness of $\bar{S}$ with respect to $\mathcal{C}$ can be defined as a cumulative probability distribution over a random variable t indicating the time duration after a change event happened in $\mathcal{C}.O$. Timeliness takes values in $[0, 1]$ for $t \in R^+$ and measures the probability of a change being captured in $F_{\mathcal{C}}(\bar{S})$ with a delay of t time units. Higher timeliness values for smaller values of t correspond to sources that get updated more frequently. A per-domain-point timeliness distribution can be defined for each domain point in $\mathcal{C}.D$.

Position bias. The position bias of $\bar{S}$ with respect to $\mathcal{C}$ measures how positive or negative the sentiments of $\bar{S}$ are towards entities contained in the domain $\mathcal{C}.D$ of the c-cluster. Let V denote a discretization of the possible sentiments (e.g., positive, negative, neutral). Given V, the position bias can be defined as a collection of $|V|$ conditional probability distributions over elements of the entity set of $\mathcal{C}$ given they are covered by $\bar{S}$. Given a sentiment $v \in V$, the probability value for v (e.g., positive) corresponds to the probability that $\bar{S}$ has sentiment v towards the data entries corresponding to $\mathcal{C}.O$ and $F_{\mathcal{C}}(\bar{S})$. The sentiment of $\bar{S}$ for a single data entry $e \in F_{\mathcal{C}}(\bar{S})$ can be extracted using standard sentiment analysis techniques [25]. The sentiments over all entries in $F_{\mathcal{C}}(\bar{S})$ can be aggregated to form the final value of the above distributions.

Nevertheless, the content of the c-cluster is unknown and only partial information is available via content samples from the sources. Given this, combining the available source samples to extract a sufficient view of the data in the cluster poses an important challenge.

2.2.3 Computing the Quality of Subsets of Sources

The third challenge is that one should be able to compute the aforementioned quality metrics efficiently for any set of sources. Computing the quality of every possible subset of sources in ad-

vance is obviously prohibitive. For certain cases [12, 29], one can estimate the overall quality for any set of sources by building offline quality profiles for each individual source and then combining those during source selection to estimate the overall quality for an arbitrary set of sources. The high-level intuition behind this approach is that all quality metrics are associated with a probability distribution (i.e., a multinomial distribution for coverage and accuracy and an empirical distribution for timeliness). If the sources are assumed to be independent the corresponding random variables are also independent, and hence, the probabilities corresponding to the quality of a set of sources can be computed efficiently using the decomposable disjunction formula. For example, given two sources S_1 and S_2 with individual coverages $C(S_1) = 0.6$ and $C(S_2) = 0.7$, the overall coverage of the integration result for S_1 and S_2 corresponds to the probability that an item from $\mathcal{C}.O$ is either covered by S_1 or covered by S_2 and is $C(S_1, S_2) = 1 - (1 - 0.6)(1 - 0.7) = 0.88$.

In reality, sources are far from independent [5, 10, 28], as they exhibit overlaps, copying relationships and/or may contradict each other. These relationships make the quality random variables for the sources dependent. Estimating the previous quality metrics in the presence of dependencies poses a major challenge as it requires: (i) extracting the source dependencies from available source samples and (ii) devising efficient techniques for computing the probability of the overall quality random variables during query evaluation. This requires performing joint probabilistic inference over the probability distributions corresponding to the different metrics.

Finally, the content of sources may be changing over time [29]. While there are cases where the quality of sources is stable over time (e.g., New York Times is a highly accurate newspaper), there might be cases where both the quality of sources and their content focus may change significantly (e.g., a blog on consumer reports may reduce its publishing rate and lose part of its credibility). Furthermore, new sources may become available or existing sources may disappear from the system. These changes give rise to a couple of important challenges. The first one is that of updating the source quality profiles and the learned source dependencies efficiently in an online fashion. The second one is being able to identify how the quality of sources changes if their updates are incorporated at different frequencies and reason about the implications of this on source selection [29].

2.2.4 Interactive Exploration

As mentioned above, users should be able to specify a certain budget for their application as well as a collection of relevant quality metrics. The budget constraint can correspond either to a monetary budget, limiting the amount of data that can be acquired, or a budget on the number of sources that a source selection solution should contain. The latter is particularly useful when a user wants to verify the content of sources manually.

While budget constraints may be natural to users, it might be hard for them to know in advance which of the quality metrics is more important or what are the possible trade-offs among these metrics due to the interdependencies between them. We expect that not even expert users know the exact data quality requirements of a task. Consider a political scientist analyzing newspaper articles to forecast interesting events. Imposing a constraint that data sources have zero delay at reporting certain events may reduce the coverage of the integration result. Moreover, users may not know the feasible level of quality given their budget and the trade-offs between the solutions that focus on maximizing the different quality metrics in isolation. For example, selecting sources that are optimal for accuracy may lead to low coverage.

Instead of optimizing the profit of integration with respect to a single quality metric or adopting a predefined weighting across metrics, source selection should be viewed as a multi-objective optimization problem with each quality metric being a separate objective. In multi-objective optimization there is usually no single feasible solution that is the optimum for all the objective functions simultaneously. Therefore, attention is paid only to the *Pareto optimal solutions*, i.e., solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives. The set of Pareto optimal solutions is called the *Pareto front*.

The user should be able to explore the different solutions on the Pareto front to identify which solution suits her task the best. As a result, a major challenge for a data source management system is to guide the user through the different source selection solutions that satisfy the user's budget and help her understand the trade-off between the integration quality achieved by different solutions. We argue that this is feasible only through an interactive process where the user will be able to explore the feasible solution space following suggestions of the data source management system. This will enable users to understand the interdependencies between the different quality metrics with respect to their integration task and identify the particular solution that suits their application. The latter raises the following challenges: (i) how can a user explore the solution space efficiently, (ii) how can a source management system present the quality profile for a set of sources in a concise and meaningful way, (iii) what are the right hints that the system should present to the user to facilitate the exploration of the solution space, (iv) how can the system take advantage of user feedback to guide the user in her interactive exploration of the solution space.

3. A PROPOSED SYSTEM DESIGN

In this section we propose a preliminary design that aims to instantiate the source analysis and query engine modules of the architecture proposed above and address the corresponding challenges. As demonstrated, one of the major challenges is reasoning about the content of sources focusing on diverse data domains. We propose using an *ontology* organized as a graph (e.g., Google's Knowledge Graph [31]) as a global relaxed schema for describing arbitrary data domains. Knowledge bases are examples of ontologies and in the remainder of the paper we will use the term knowledge base to refer to ontologies. A knowledge base acts as an information repository that provides a means for information to be collected, organized, shared, searched, and utilized. A knowledge base can be viewed as a collection of ***facts*** (or *instances*) that describe information about entities and their properties, and ***concepts*** that describe information about entity types and their properties. Both facts and concepts can be represented as nodes of the knowledge base. Given a knowledge base, a *context cluster* (c-cluster) (e.g., "Sports in the USA") can be described as a collection of concepts and/or entities. Different sources may focus on different c-clusters. For example, "The Economist" mentions data from the c-cluster of "economy" but a Web table containing the populations of countries across the globe corresponds to a different c-cluster. Notice that both c-clusters can be represented using a collection of concepts and/or entities from a knowledge base. Next, we describe how one can build the source analysis module and the query engine module (Figure 1) around a knowledge base.

3.1 Source Analysis Module

To reason about the content of different data sources and their data quality we propose augmenting the knowledge base with a *correspondence graph*. Specifically, the nodes in the correspondence graph are either data sources (referred to as *source nodes*)

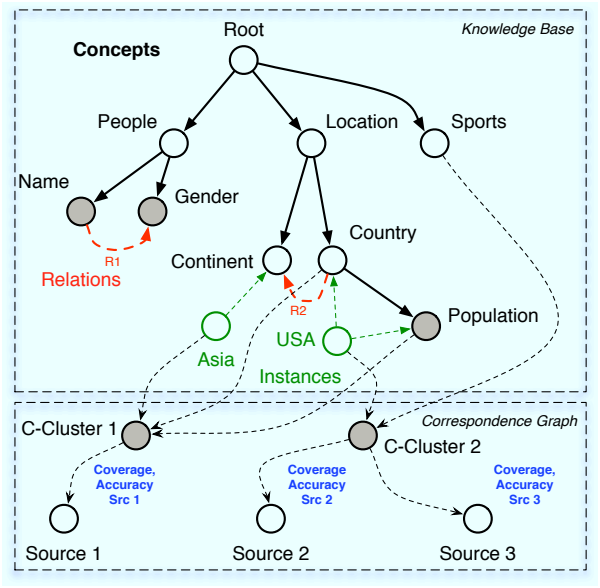

Figure 2: An example of a knowledge base and a correspondence graph with two c-cluster nodes corresponding to the population of countries in Asia and sports in the USA.

or c-clusters of concepts and/or entities as dictated by the available sources (called *c-cluster nodes*). The edges in the correspondence graph connect each source node with c-cluster nodes and c-cluster nodes with the corresponding concepts and entities in the knowledge base. Each edge from a source to a c-cluster node is annotated with a quality profile of that source for that specific c-cluster, and each c-cluster node is associated with local information about the dependencies of the data sources that are connected to it. An example of a knowledge base and a correspondence graph is shown in Figure 2. There are two c-cluster nodes, one corresponding to the population of countries in Asia and one to sports in the USA.

We describe a preliminary approach for constructing the correspondence graph. We propose a two step approach where we first learn the latent c-cluster nodes and then compute the quality profiles and data source dependencies for each c-cluster node.

Step 1. The c-cluster nodes in the correspondence graph can be viewed as a content-based clustering of the available data sources. Furthermore, each of these nodes is associated with a collection of concepts and/or instances of the knowledge base. The following approach can be used to construct these nodes. Each source can be viewed as a *collection of entries*, where each entry is a *conjunction* over concepts and/or instances. To obtain this representation, we must annotate the content of each source with concept and instance labels from the knowledge graph. Several techniques have been proposed for obtaining these annotations [1, 24]. Once the content of sources is represented as a collection of concept and instance conjunctions, one can use a *mixed membership model* [6] to describe how the content of sources is generated considering the c-cluster nodes. Each source is modeled as a mixture over the c-cluster nodes. The c-cluster nodes are shared across all sources but the mixture proportions vary from source to source (they can also be zero). Each c-cluster node describes a distribution over concepts or events. We plan on building upon recent work on sparsity inducing non-parametric latent variable learning techniques [13, 3]. Sparsity is necessary as each c-cluster node should contain only a limited number of concepts and instances.

Step 2. After discovering the c-cluster nodes, we can compute the quality of each source with respect to each c-cluster node it is connected to. To do the latter, we need to collectively analyze the content of all sources connected to a c-cluster node. We propose following an approach similar to Rekatsinas et al. [29] where samples from all the sources are integrated into a single dataset forming the content of the c-cluster and then each individual sample is compared with the integrated data to compute the source quality.

Apart from the individual source quality profiles, we also need to learn the quality dependencies across sources. Recall that the quality metrics presented in Section 2.2.2 can be expressed as probability distributions. When sources are dependent, the random variables corresponding to their quality distributions are dependent. These dependencies can be extracted from the available source samples. We conjecture that these dependencies can be represented using a *factor graph* [20], i.e., a particular type of graphical model that enables efficient computation of marginal distributions, over the source random variables. We plan to explore how structure learning techniques from the statistical relational learning literature [15] can be used to solve this problem. These factor graphs will also enable computing the quality of an arbitrary set of sources via probabilistic inference. The latter is necessary for solving the problem of source selection during query time as we describe next.

3.2 Query Engine

Queries in the context of a data source management system correspond to descriptions of the user's integration requirements. We envision a system where queries are free-text descriptions containing references to multiple entities and concepts. Part of the query will correspond to specifying an integration budget constraint either in terms of the maximum amount of money to spend for acquiring data or the maximum number of sources to be used for the task. Finally, the user will have the capability of selecting which quality metrics are relevant to her integration task. Given the user requirements as input the query engine should perform the following steps: (i) analyze the description of the integration task and reason about its semantic content by mapping concept or entity mentions to the knowledge base, (ii) identify the relevant c-cluster nodes in the correspondence graph and retrieve the sources that are relevant to the user's input, (iii) find Pareto optimal source selection solutions considering the quality and cost specifications of the user, (iv) present these solutions to the user and allow the interactive exploration of the retrieved results.

Semantic Analysis. The concepts or entities mentioned in the query can vary significantly. Queries can also be ambiguous (e.g., Apple the company versus apple the fruit). To handle such cases a data source management system should be able to reason about the semantic content of user descriptions. Techniques from keyword search over knowledge bases applied to lists [26] or web-tables [9] can be extended to support these needs. Once the query concepts and entities are retrieved, the query engine needs to identify the relevant c-cluster nodes. Following the mixed membership model described above, we can consider the query as a collection of concepts and instances and find its mixture proportions with respect to the c-cluster nodes in the correspondence graph. Inferring the mixture proportions can be done using approaches similar to the ones introduce by Blei et al. [6]. Once the mixture proportions are known, we can identify the sources that are relevant to each of the c-cluster nodes having a non-zero mixture proportion for the query by traversing the correspondence graph. To identify the set of valuable sources for the given query we can solve the problem of source selection [12, 29]. The benefit of integration can be described as a linear combination of the integration quality of each individual c-cluster node using the mixture proportions as weights.

Pareto Optimal Answers. Source selection identifies the optimal

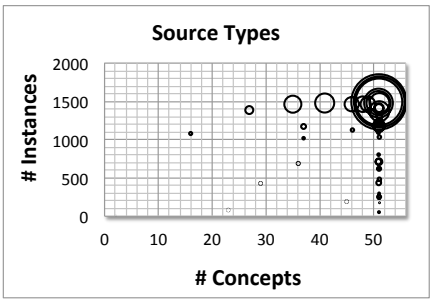

Figure 3: An illustrative bubble chart describing the sources of a potential solution to a user query.

set of sources to be used for a specific integration task by trying to maximize the profit of integration with respect to any budget constraints. As mentioned in the previous section, source selection should be cast as a multi-objective optimization problem and the query engine should be able to find the set of Pareto optimal solutions. Discovering all the solutions on the Pareto front might be expensive, thus, efficient approximation and exploration techniques have been proposed in the optimization literature [34]. Moreover, algorithms for computing the Pareto frontier of a finite set of alternatives have been studied in the skyline query literature [16, 21].

Interactive Exploration. We propose a two-level visualization approach for exploring solutions on the Pareto front. We argue that the query engine system should return a ranked list with *diverse* solutions on the Pareto frontier and mention the quality characteristics for the selected set of sources, the number of sources and the total integration cost. If the user selects to explore the content of the solution the system will present a bubble chart with all the sources in the solution. The dimensions of the bubble chart should characterize the content of each source while the size of the bubble should correspond to the actual size (with respect to number of entries) of each source. We argue that the following dimensions are necessary to describe each source: (i) the *concept focus* of the source, i.e., number of different concepts mentioned in the source, and (ii) the *instance focus* of the source, i.e., the number of different instances mentioned in the source. If the user selects a specific bubble from the bubble chart, details regarding the name and quality of the sources should be presented to the user. This information can be directly retrieved from the correspondence graph and does not need to be computed during query time. An example of such a bubble chart is shown in Figure 3. Finally, we envision a system that will provide the user with the capability of exploring the neighborhood of a solution from the initial list. This can be done either (i) by removing sources from a running solution or (ii) by recommending solutions in the Pareto frontier neighborhood of the running solution. These functionalities require reasoning about the distance of solutions on the Pareto frontier introducing a new challenge.

4. SYSTEM DEMONSTRATION

Here, we describe a demo proposal for our data source management prototype. The main focus of the data source management system introduced before is to enable users to discover and explore valuable sets of sources for diverse integration tasks. In this way, the demo focuses on exposing these functionalities to the audience.

Set-up and Scenarios. The main idea for the demo is that the audience will get direct access to a prototype of our data source management system. We will provide access to our system via a web-interface for this purpose. The data and the system will be stored and running on a remote server, however, the audience will have the opportunity to explore the internal source indexing mechanism of our system and issue source selection queries against it.

For the purpose of the demonstration we will focus on data extracted from the Global Database of Events, Languages and Tone (GDELT) [22]. GDELT is a repository that monitors news media from all over the world and extracts geo-referenced records that correspond to different events and interactions between diverse groups of people, international organizations, countries etc. GDELT gets updated every day with new event extractions. This repository is rather prominent amongst political scientists and data mining scientists as they use it to find supporting evidence for new theories and validate new techniques for forecasting events of interest [30].

We believe that GDELT is the right fit to demonstrate the usefulness and practicality of our data source management system due to the large number of data sources available, the available daily updates and the heterogeneity that sources exhibit both with respect to their content and their quality. In our recent work on source selection [29], we studied a snapshot of GDELT over a period of one month from January 2014 to February 2014 and have evaluated the effectiveness of source selection. That snapshot contained 15,275 news sources providing 2,219,704 distinct events corresponding to 242 different locations and 236 different event concepts. We refer the reader to Rekatsinas et al. [29] for a more detailed description of the data set. For this demo we plan to use a recent larger snapshot of GDELT including all sources contained in the dataset. We aim to enable daily updates in our data source management system.

Correspondence Graph Exploration. During the first part of the demonstration, users will be able to explore the correspondence graph part of our system. In particular, we will provide visualizations illustrating a fixed set of source nodes together with their corresponding c-cluster nodes and quality summaries. Moreover, the users will have the capability of selecting a specific c-cluster node and see the different concepts and entities connected with it.

Source Selection Queries. For the second part of the demonstration, we will provide the users with a set of example queries they can execute against our system. The users will have the opportunity to explore the solutions for these queries using the techniques introduced in Section 3.2. Our goal is for users to understand the trade-offs between different quality metrics (including coverage, timeliness and accuracy) of sources in GDELT. Users will also have the opportunity to issue their own free-text queries.

Summary. Overall, the demo will allow users to: (i) understand the internal source indexing mechanism (i.e., the correspondence graph) of our prototype system, (ii) issue queries against it and (iii) explore the corresponding source selection solutions via a web-interface. Users will need to play the role of a political scientist and use our system to discover the most valuable sources for their own analysis applications.

5. RELATED WORK

Prior work mainly focuses on isolated aspects of data source management, and to our knowledge, there has been no systematic approach to developing a source management system over large numbers of data sources. There is much work on schema mapping and semantic integration of different sources [8, 33, 18]. This line of work focuses on the construction of a global schema or a knowledge base describing the domain of the data sources, and its final goal is not reasoning about the content and quality of sources. Moreover, most of that work focuses on sources from a specific domain and does not present results for largely heterogeneous sources. Web table search [8, 24, 9, 35, 14] is also closely related to data source search. Most of the proposed techniques consider user queries

and return tables related to specific keywords present in the query. However, the keyword-based techniques fail to capture the semantics of natural language, i.e., the intentions of the users, and thus they can only go as far as giving relevant hits. Using the knowledge base as the entry point of data source search will enable us to clearly capture the intentions of the user and return more useful results. Further, extending data source search to recommend sets of sources to be integrated and analyzed collectively, as we propose to do, is a useful functionality in many domains (e.g., data driven journalism) where users are not experts and want an efficient way of exploring multiple data sources. Apart from Web table search, more generic data search systems, such as Microsoft's Power BI [2], have been recently proposed. Nonetheless, such systems focus on facilitating data integration and do not provide analysts with the functionality to understand the quality of data sources and the trade-off between different quality metrics. Finally, our work on source selection [12, 29] has considered problems where all sources follow a common schema and focus on a single data domain.

6. CONCLUSIONS

In this paper, we presented our vision for a data source management system that will enable users to discover the most valuable sources for their applications. Given a user's budget our system also enables the interactive exploration of different sets of sources allowing the user to truly understand the quality and cost trade-off between different integration options. We discussed the major challenges in building such a system such as, supporting diverse integration tasks and multiple users, assessing the quality of data sources and enabling the interactive exploration over different sets of sources. We presented a preliminary design of such a system addressing these challenges. We believe that it is the time for a new type of data portals that will allow data scientists and analysts to find the most valuable data sets for their tasks and limit the person-hours spent in validating the quality of data.

Acknowledgements

The authors would like to deeply thank Amol Deshpande for the many fruitful and insightful discussions on the subject as well as his help on proofreading the manuscript.

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

7. REFERENCES

[1] DBpedia spotlight. https://github.com/dbpedia-spotlight/.

[2] Excel Power BI. http://www.microsoft.com/en-us/powerbi.

[3] R. Balasubramanyan and W. W. Cohen. Regularization of latent variable models to obtain sparsity. In *SDM*, 2013.

[4] M. Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. *PVLDB*, 4, 2011.

[5] L. Berti-Equille, A. D. Sarma, X. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*, 2009.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, Mar. 2003.

[7] M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Extraction and integration of partially overlapping web sources. *PVLDB*, 3, 2013.

[8] M. J. Cafarella, A. Halevy, and N. Khoussainova. Data integration for the relational web. *PVLDB*, 2009.

[9] A. Das Sarma, L. Fang, N. Gupta, A. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu. Finding related tables. *SIGMOD*, 2012.

[10] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. *PVLDB*, 2, 2009.

[11] X. L. Dong, A. Halevy, and C. Yu. Data integration with uncertainty. *The VLDB Journal*, Apr. 2009.

[12] X. L. Dong, B. Saha, and D. Srivastava. Less is more: selecting sources wisely for integration. *PVLDB*, 2013.

[13] G. Elidan and N. Friedman. Learning Hidden Variable Networks: The Information Bottleneck Approach. *J. Mach. Learn. Res.*, 2005.

[14] J. Fan, M. Lu, B. C. Ooi, W.-C. Tan, and M. Zhang. A hybrid machine-crowdsourcing system for matching web tables. In *ICDE*, 2014.

[15] L. Getoor and B. Taskar. *Probabilistic Relational Models*. The MIT Press, 2007.

[16] P. Godfrey, R. Shipley, and J. Gryz. Algorithms and analyses for maximal vector computation. *The VLDB Journal*,16(1), 2007.

[17] H. Gonzalez, A. Halevy, C.S. Jensen, A. Langen, J. Madhavan, R. Shapley, and W. Shen. Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud. In *SoCC*, 2010.

[18] O. Hassanzadeh, K. Q. Pu, S. H. Yeganeh, R. J. Miller, L. Popa, M. A. Hernández, and H. Ho. Discovering linkage points over web data. *PVLDB*, 6, 2013.

[19] Y. Keneshloo, J. Cadena, G. Korkmaz, and N. Ramakrishnan. Detecting and Forecasting Domestic Political Crises: A Graph-based Approach. *WebSci*, 2014.

[20] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[21] D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. *VLDB*, 2002.

[22] K. Leetaru and P. Schrodt. Gdelt: Global data on events, language, and tone, 1979-2012. *Inter. Studies Association Annual Conf.*, 2013.

[23] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: is the problem solved? *VLDB*, 2013.

[24] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *PVLDB*, 3, 2010.

[25] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. *ACL*, 2002.

[26] R. Pimplikar and S. Sarawagi. Answering table queries on the web using column keywords. *PVLDB*, 5, 2012.

[27] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Commun. ACM*, 45(4), 2002.

[28] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. *SIGMOD*, 2014.

[29] T. Rekatsinas, X. L. Dong, and D. Srivastava. Characterizing and selecting fresh data sources. *SIGMOD*, 2014.

[30] S. Schutte and K. Donnay. Matched wake analysis: finding causal relationships in spatiotemporal event data. *Political Geography*, 41, 2014.

[31] A. Singhal. Introducing the knowledge graph: Things, not strings. Official Blof (of Google), 2012.

[32] P. Upadhyaya, M. Halevy, M. Balazinska, D. Suciu, W. Shen, H. Hacigumus. Affordable Analytics on Expensive Data. *Data4U*, 2014.

[33] P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. *PVLDB*, 4, 2011.

[34] B. Wilson, D. Cappelleri, T. W. Simpson, and M. Frecker. Efficient Pareto Frontier Exploration using Surrogate Approximations. *Optimization and Engineering*, 2, 2001.

[35] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. *SIGMOD*, 2012.