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

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

Amol Deshpande
University of Maryland
amol@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, leading to a rapid increase in the number of open-access data sources and services, such as cloud-based data markets and data portals, that facilitate the collection, publishing and trading of data. Data sources are typically heterogeneous in their content, the quality of data they provide, and the fees they may require for accessing their entries. However, when the number of data sources is large, humans have a limited capability of reasoning about the actual quality of sources and the trade-off between the benefits and costs of acquiring and integrating sources. Given the sheer number of available data sources, analysts must (1) identify sources that are relevant to their integration task, (2) discover sources that potentially satisfy the quality and budget requirements of their applications with few effective clues about the quality of the sources, (3) repeatedly invest many man-hours in assessing the eventual usefulness of data sources. All three steps require investigating the content of sources manually, integrating subsets of them and evaluating the actual benefit of the integration result for a desired application. In this paper we explore the problems of appraising the quality of data sources and identifying the most beneficial sources for diverse integration tasks. 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 discovery of valuable sources for user specified integration tasks. We argue that the proposed system can dramatically ease the *Discover-Appraise-Evaluate* interaction loop that many analysts follow today to discover beneficial sources for their tasks.

1. INTRODUCTION

In the last few years, the number of data sources available for integration and analysis has risen 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), 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 to acquire and analyze data from multiple data sources.

However, the sheer number of available data sources makes it challenging for a user to identify sources that are truly beneficial to her integration or analysis task. First, many sources provide erroneous or stale data entries that can be detrimental to the quality of integration [11, 19]. Second, sources may provide duplicate and redundant data, making hard to identify the unique information a source is providing [7, 19]. Third, sources exhibit significant heterogeneity in representation of stored data (e.g., the schema they follow), making it challenging for a user to discover all relevant sources for her integration task [9]. Finally, acquiring and integrating data comes with a monetary and computational cost, and hence, integrating every available source may not be worthwhile or even feasible due to budget constraints. These challenges give rise to the natural questions of (i) how can one discover the value of data in a rigorous fashion and (ii) how can one identify the most beneficial data sources for arbitrary integration tasks.

Recent work [12, 21] showed that given a fixed data domain, the benefit of integration can be specified using rigorous quality metrics such as *coverage*, *accuracy* and *freshness*. Moreover, this work introduced the paradigm of *source selection* to reason about the benefits and costs of acquiring and integrating data. Yet the proposed techniques focus on pre-defined integration tasks and do not provide support for a diverse set of users to specify their integration task. Moreover, this approach requires that a user knows exactly which quality metric is of importance to her or what her desired trade-off between multiple metrics is. This last requirement is rather unrealistic as users rarely know what is the right trade-off between different quality metrics.

In this paper we introduce our vision for a *source management system* that will enable users to specify heterogeneous integration tasks as keyword queries and allow them to interactively discover the most important sources for their specified task. Given an extensive collection of rigorous data quality metrics, we envision a system that automatically discovers the quality of different data sources and provides source selection capabilities to enable discovery and exploration of data sources. A key characteristic of the proposed system is to help users understand which quality metrics are important for their integration task and enable them to discover sources whose integration result will maximize the desired metrics under any specified budget constraints.

The remainder of the paper is organized as follows. In Section 2 we discuss the key requirements and challenges in building a data source management system, including defining and computing multiple quality metrics that characterize the content of sources,

supporting diverse integration tasks expressed as keyword queries, and enabling the interactive exploration of data sources. Then, Section 3 presents an overview of our proposed system, introduces the different modules of the system and proposed techniques for addressing the aforementioned challenges. Finally, Section 4 discusses related work and Section 5 concludes the paper.

2. DISCOVERING VALUABLE SOURCES

In this section we describe two tasks that require integrating data from multiple sources. Our goal is to identify common characteristics across these tasks and introduce the main operations that a source management system should support to enable the discovery of valuable sources. Finally we describe the challenges in each of these operations.

2.1 Motivating Examples

Data from multiple sources can be combined to perform diverse tasks. Next, we present two such tasks and identify the main elements needed to describe an integration task. The first task we consider corresponds to combining data from multiple news-papers to validate a new theory in political sciences, while the second one corresponds to analyzing social media activity to spot emerging trends used for companies, news organizations and financial institutions.

Scenario 1. We consider the Global Database of Events, Languages and Tone (GDELT) [18] where news articles from thousands of news domains (sources) are aggregated in a single repository. The articles are analyzed and events, defined as pairwise interactions between actors, are extracted from them. Actors typically correspond to well-known organizations, including countries and international organizations. The extracted events are quite diverse ranging from economic agreements between companies to violent acts between parties and are associated with certain locations, actors (corresponding to organizations), and a description containing the political views of the actors and a characterization of the event. Such a repository can be used for diverse analytics tasks. We focus on a scenario where analysts integrate events mentioned in a diverse set of news media sources and analyze them collectively to detect patterns and evaluate new theories. In particular, we focus on the work by Schutte et al. [22]. In this work, the authors focus on finding causal relationships in spatiotemporal event data and use data from GDELT to build test cases for validating their theory. In particular, they are interested in examining how the civilian assistance to US forces changed in response to indiscriminate insurgent violence in Iraq. They focus on a specific time-window from 2003 to 2010, consider events in specific locations in Iraq and consider actors corresponding to specific ethnic groups in Iraq.

Scenario 2. Next, we consider the company TrendSpottr [3] and its partnership with Datasift [1]. Datasift is a company that extracts social media data from multiple sources, including Twitter, Facebook, Blogs etc., and offers a common API for accessing dozens of different data sources in real-time. TrendSpottr analyzes real-time data streams such as Twitter and Facebook to spot emerging trends at their earliest acceleration point. They require access to historical data, demographic information and geo location information as all of this data is important to the predictive model used by the company. Typical tasks supported by TrendSporttr are viral news discovery corresponding to different locations and specific organizations, brand and reputation monitoring, ranging from company acquisitions to sports, market predictions and many others.

In both scenarios, we see that specifying an integration task involves providing information about its *location*, a set of exact instances or types of *organizations* or *people* involved in it and a *time window* associated with the task (this can be either a real-time task or a historical task over a past time-window). Moreover, it is easy to see that different types of data quality are implicitly relevant to each of the aforementioned tasks. For example, *completeness* is important for the first task but timeless is not crucial as delayed mentions of events will not affect the quality of the analysis. On the other hand, *timeliness* is very important in the second application. Analyzing sources that exhibit significant delays beats the purpose of the application which is early detection of trends.

2.2 Key Operations

Building upon the previous examples, we present the main operations that a source management system should support for discovering valuable sources. First, we consider the input to such a system. Typically a user would start by providing the following information about the data integration task and the desired data sources: (i) a free-text description of the integration task corresponding to a collection of mentions to either specific objects or types (concepts) of locations, organizations and people, (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 as input a source management system should perform the following operations:

1. **Discover.** Given the description of the integration task 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 to the integration task, the system should automatically find subsets of sources that if integrated together maximize the quality of integrated data under the imposed budget constraints by the user. The system should identify multiple solutions that correspond to maximizing different quality metrics. This is necessary to diversify the solutions presented to the user in the next phase. Exploring different solutions is necessary for the user to understand the trade-offs between different quality metrics and her budget constraints, thus, identifying the solution that is best suited for her task.

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 explore the neighborhood of a solution by removing sources or examining solutions with similar characteristics that contain different sources.

2.3 Challenges

Next, we discuss the main challenges in each of the operations presented above.

2.3.1 *Diverse Integration Tasks*

The first challenge is supporting user queries over heterogeneous data domains. As illustrated in the previous examples, the different concepts or entities associated with an integration task can vary significantly. Moreover, the data from the sources can be both structured and unstructured. Therefore, a data source management system should be able to reason about the semantic content of different types of data ranging from tables to free-text. Moreover, the system should be able to semantically match the content of the user-provided task description to the content of the available sources included in the system. To fulfill these requirements one should

be able to support scalable search over different types of data (e.g., structured and unstructured) while following an open-domain assumption and not restricting user queries to a pre-specified vocabulary.

2.3.2 Data Source Quality

The second challenge is defining the quality of data and being able to assess the quality of different sources in an automated way. Traditionally, the quality of data sources had been measured using the amount of erroneous information provided by the source. However, as discussed earlier there are multiple quality metrics that might be of interest to a user.

Given an integration task $\mathcal{I}$ let its data universe $\mathcal{U}(I)$ to be the set of all objects that are relevant to task $\mathcal{I}$. Notice, that $\mathcal{U}(I)$ is not fully known but only partial information is available for this universe via the data sources. We assume that each object $r \in \mathcal{U}(I)$ has a set of attribute values denoted by $r.A$. For example these attributes can correspond to a location or a type or instance of an organization as described in the previous examples. Moreover, we assume that either the objects in $\mathcal{U}(I)$ or their values may change over time. Given a set of sources S relevant to task I and an integration model F we have that $F(S) \subseteq \mathcal{U}(I)$, where $F(S)$ denotes the set of objects extracted after integrating all sources in S. Using this notation we define the following metrics:

Coverage. The coverage of S with respect to $\mathcal{U}(I)$ can be defined as the probability that an object chosen from $\mathcal{U}(I)$ uniformly at random will be present in $F(S)$. Notice that we do not require that the attributes of the object $r.A$ are correct.

Accuracy. The accuracy of S with respect to $\mathcal{U}(I)$ can be defined as the probability that an object chosen from $F(S)$ uniformly at random is correct with respect to $\mathcal{U}(I)$. The latter means that the object must be present in $\mathcal{U}(I)$ and all its attributes should have the correct values capturing errors due to both noisy and stale data.

Timeliness. The timeliness of S with respect to $\mathcal{U}(I)$ can be defined as the cumulative probability distribution of a change in the objects of $\mathcal{U}(I)$ being reflected in $F(S)$ with a delay of at most t time units. Notice that this last quality metric does not correspond to a single number but an entire distribution characterizing the delays the source exhibits.

Publication bias. The publication bias of S with respect to $\mathcal{U}(I)$ can be defined as the selective revealing or suppression of data entries from $\mathcal{U}(I)$. More formally we can measure the publication bias of S by analyzing the publishing probability of S for all data entries in $\mathcal{U}(I)$. The higher the mass of this distribution is concentrated over a specific subset of entries from $\mathcal{U}(I)$ the higher the bias of the source should be. The bias of a source can be quantified using the entropy of the source publishing distribution.

Notice that all definitions presented above, compare the content of a source or a set of sources with the data universe corresponding to an integration task. However, the actual content of the universe is unknown and only partial information is available via content samples from the sources. Given this, the main challenge becomes combining the available source samples to extract a sufficient view of the data universe. Another challenge is that one should be able to compute the aforementioned quality metrics efficiently for any set of sources. Computing the quality of any possible set of sources in advance is obviously prohibitive. However, for certain cases [12, 21], 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 each of the aforementioned quality metrics is associated with a random variable following a specific probability distribution (i.e., a Bernoulli 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, the overall coverage for a set of sources S_1 and S_2 with individual coverages $C(S_1) = 0.6$ and $C(S_2) = 0.7$ corresponds to the probability that an item from $\mathcal{U}(I)$ 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, however, sources are far from independent [5, 10], as they exhibit overlaps, copying relationships and/or may contradict each other. These relationships make the quality random variables for each source dependent. Estimating the aforementioned quality metrics under the presence of dependencies imposes a major challenge as (i) it requires 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. In fact, the latter often corresponds to performing probabilistic inference over a joint distribution formed by the quality random variables corresponding to the desired set of sources.

2.3.3 Interactive Evaluation

As mentioned above, users should be able to specify a certain budget for their integration task. This can be either 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 specifying a constraint on the integration budget is natural to users, specifying a constraint on the data quality with respect to the aforementioned metrics is rather intricate due to the interdependencies between the different metrics and we expect that not even expert users know the exact data quality requirements of a task. Consider for example a political scientist analyzing news papers articles to forecast interesting events. Imposing a constraint on using only data sources that have exhibited zero delay with respect to certain events may reduce the accuracy of the integration result. Moreover, users may not know what is the feasible level of quality given their budget and what are the trade-off between solutions that focus on maximizing the different quality metrics in isolation. For example, selecting sources optimizing for accuracy may lead to an integration result of low coverage.

So a major challenge for a 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 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, and (iii) what are the right hints that the system should present to the user to facilitate the exploration of the solution space.

3. SYSTEM OVERVIEW

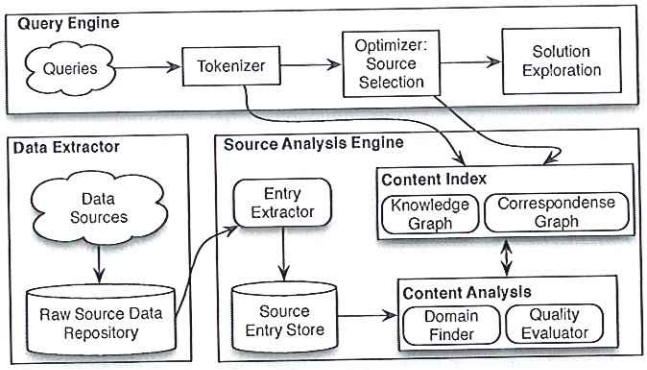

Figure 1: Source Management System Architecture.

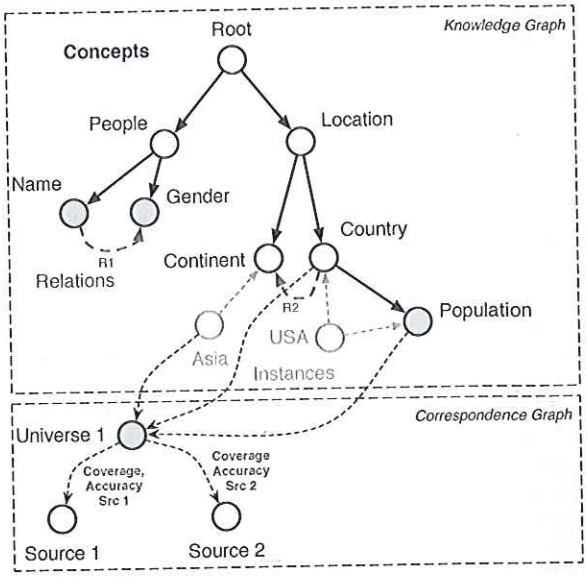

Figure 2: An example of a knowledge graph and a correspondence graph with a universe node corresponding to the population of countries in Asia.

In this section we propose a preliminary design that aims to address the challenges specified in the previous section. An overview of the proposed architecture for a data source management system is shown in Figure 1. The system is composed of (i) a data extractor module, responsible for collecting the raw data from different sources, (ii) a source analysis engine and (iii) a query engine. The source analysis engine analyzes the content of sources, evaluates their quality with respect to the quality metrics introduced in the previous section and keeps an index describing both the content and the quality of each source. The query engine is given a user query (i.e., a description of the desired integration task) as input. Using this input it first identifies the data universe corresponding to the query, and then, the set relevant sources. Next, it detects the most valuable sources for integration using source selection and presents the results to the user while enabling interactive exploration. Next, we focus on the source analysis engine and the query engine and discuss how these two components address the challenges and requirements presented above. We omit the first module of extracting data from different sources as it does not entail any technical challenges.

3.1 Source Analysis Module

A data source management system should be able to support sources covering completely heterogeneous data universes. For example, it should be able to analyze the content of a structured Web table providing financial data and the unstructured content of news articles extracted from a news paper. This is necessary for diverse integration tasks as described in Section 2.3.1.

We propose using a *knowledge graph*, i.e., a *knowledge base* organized as a graph (e.g., Google Knowledge Graph), as a global relaxed schema for describing arbitrary data universes. 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* 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 graph. Given a knowledge graph, a data universe (e.g., "Sports in the USA") can be described as a collection of concepts and/or entities. To reason about the different data universes the available sources cover and the source quality we propose augmenting the knowledge graph with a *correspondence graph*.

Specifically, the correspondence graph contains data sources as nodes (referred to as *source nodes*), a set of nodes corresponding to clusters of concepts and/or entities as dictated by the available sources (referred to as *universe nodes*), and connects each source node with the universe node corresponding to the concepts and facts covered by that source. The universe nodes are also connected with the corresponding concepts and entities. Each edge from a source to a universe node is annotated with a quality profile of that source for that specific data universe, and each universe node is associated with local information about the dependencies of the data sources that are connected to it. An example of a knowledge graph and a correspondence graph is shown in Figure 2. Next, we describe a preliminary approach for constructing the correspondence graph. We propose a two step approach where we first learn the universe nodes for a set of data sources and then compute the quality profiles and data source dependencies for each universe node.

The universe nodes in the correspondence graph correspond to a clustering of the content of the available data sources. Furthermore, each of these nodes is associated with a collection of concepts and/or instances of the knowledge graph. 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 [2, 20]. 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 universe nodes. Each source is modeled as a mixture over the universe nodes. The universe nodes are shared across all sources but the mixture proportions vary from source to source (they can also be zero). Each universe 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, 4]. Sparsity is necessary as each universe node should contain only a small number of concepts and instances.

After learning the universe nodes in the correspondence graph we can collectively analyze the relevant content of all sources corresponding to each node in order to extract a quality profile for

each source. In particular, we propose following an approach similar to Rekatsinas et al. [21] where samples from all the sources are integrated into a single dataset corresponding to the content of the data universe and then each individual sample is compared with the integrated data to extract the source quality profile.

Apart from the individual source quality profiles we also need to learn the quality dependencies across different sources. Recall that the quality metrics presented in Section 2.3.2 can be expressed as probabilities corresponding to a distribution associated with source-quality random variables. In fact when sources are dependent these random variables are dependent, and hence, these dependencies need to be extracted from the available source samples. We conjecture that these dependencies can be represented using a *factor graph*, 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 sources selection during query time as we describe next.

3.2 Query Engine

Queries against a data source management system correspond to descriptions of an integration or analytics task. We envision a system where queries will correspond to free-text descriptions of an integration task 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 such a query the query engine of a source management system should perform the following steps: (i) first the content of the query should be mapped to the knowledge graph and the corresponding universe nodes of the correspondence graph, (ii) the universe node contains information about which sources are relevant to the task of the user and restricts the scope of available sources for performing source selection, (iii) the query engine should find the subset of sources that maximize the quality of the integration result with respect to the selected quality metrics given the specified budget constraint, (iv) instead of presenting a single answer to the user or a ranked list of answers, the query engine should allow the user to interactively explore the retrieved results. Next, we provide an overview of each of these steps.

Given the text of a query, we can use techniques similar to those mentioned before for annotating the content of sources with labels corresponding to concepts and instances of the knowledge graph. Following the mixed membership model described above, we can considering the query as a collection of concepts and instances and find its mixture proportions with respect to the universe 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 then we can identify the sources that are relevant to each of the universe 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, 21]. The benefit of integration can be described as a linear combination of the integration quality of each individual universe node using the mixture proportions as weights.

Source selection identifies the optimal set of sources to be used for a specific integration task by trying to maximize the benefit of integration with respect to any budget constraints. The benefit of integration can be expressed as a function of the quality of the integration result. However, there are multiple quality metrics and as mentioned above, it might be hard for user to know in advance which of these metrics is more important or what are the possible trade-offs among these metrics. Instead of optimizing the benefit of integration with respect to a single quality metric or adopting a predefined weighting across metrics, we propose casting source selection as a multi-objective optimization problem where each quality metric is considered as 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*. We argue that the user should be able to explore the different solutions on the Pareto front to identify which solution suits her task the best. Discovering all the solutions on the Pareto front might be expensive, thus, efficient approximation and exploration techniques have been proposed in the optimization literature [24]. Next, we propose a two-level visualization approach for exploring solutions on the Pareto front.

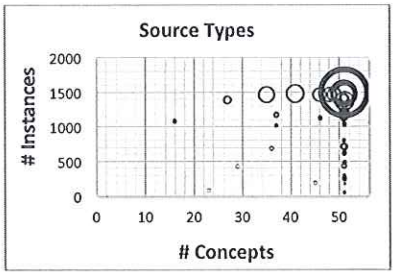

Figure 3: A fictional bubble chart describing the sources of a potential solution to a user query.

We argue that first 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 will all the sources in the solution. The dimensions of the bubble chart should be 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. Notice that this information can be directly retrieved by the correspondence graph an 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 (i) either by removing sources from a running solution by allowing the user to select the ones to be removed from the bubble chart or (ii) by recommending solutions in the Pareto frontier neighborhood of the running solution. Notice that all the above functionalities require reasoning about the distance of solutions on the Pareto frontier introducing a new challenge.

4. RELATED WORK

Most of the prior work 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 the problems of schema mapping and semantic integration of different sources [8, 23, 17, 16]. That line of work focuses on the construction of a global schema or a knowledge graph describing the domain of the data sources, and its final goal is not reasoning about the concepts that sources cover. 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, 20, 9, 25, 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 the language, i.e., the intentions of the users, and thus they can only go as far as giving relevant hits. Using the knowledge graph 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. Finally, our work on source selection [12, 21] has consider problems where all sources follow a common schema and focus on a single data universe.

5. CONCLUSIONS

In this paper, we argued that due to the vast amounts of available data sources it not beneficial for users to integrate all available sources as this can either be detrimental to the quality for the integration result or very expensive due to the fees sources charge to grant access to their data. We presented our vision for a data source management system that will enable users to not only discover the most valuable sources for their integration or analysis task given their budget but 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 including supporting fully heterogeneous integration tasks and multiple users, assessing the quality of data sources and enabling the interactive exploration over different sets of sources and presented a preliminary design of a source management system addressing these challenges. We believe that it is really about time for a new type of data portals that will allow data enthusiasts to find the most beneficial data sets for their tasks and limit the man-hours spent in validating the quality of data.

6. REFERENCES

[1] Datasift. `http://datasift.com`.

[2] Dbpedia spotlight. `https://github.com/dbpedia-spotlight/`.

[3] Trendsporttr. `http://trendspottr.com`.

[4] R. Balasubramanyan and W. W. Cohen. Regularization of latent variable models to obtain sparsity. In *SDM*, pages 414–422, 2013.

[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.*, 3:993–1022, Mar. 2003.

[7] M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Extraction and integration of partially overlapping web sources. *PVLDB*, 6(10):805–816, 2013.

[8] M. J. Cafarella, A. Halevy, and N. Khoussainova. Data integration for the relational web. *PVLDB*, 2, 2009.

[9] A. Das Sarma, L. Fang, N. Gupta, A. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu. Finding related tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 817–828. ACM, 2012.

[10] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. *Proc. VLDB Endow.*, 2(1):550–561, Aug. 2009.

[11] X. L. Dong, A. Halevy, and C. Yu. Data integration with uncertainty. *The VLDB Journal*, 18(2):469–500, Apr. 2009.

[12] X. L. Dong, B. Saha, and D. Srivastava. Less is more: selecting sources wisely for integration. In *Proceedings of the 39th international conference on Very Large Data Bases*, PVLDB'13, pages 37–48. VLDB Endowment, 2013.

[13] G. Elidan and N. Friedman. Learning Hidden Variable Networks: The Information Bottleneck Approach. *J. Mach. Learn. Res.*, 6:81–127, 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*, pages 976–987, 2014.

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

[16] 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.

[17] O. Hassanzadeh, S. H. Yeganeh, and R. J. Miller. Linking semistructured data on the web. In *WebDB*, 2011.

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

[19] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: is the problem solved? In *Proceedings of the 39th international conference on Very Large Data Bases*, PVLDB'13, pages 97–108. VLDB Endowment, 2013.

[20] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *PVLDB*, 3(1):1338–1347, 2010.

[21] T. Rekatsinas, X. L. Dong, and D. Srivastava. Characterizing and selecting fresh data sources. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 919–930. ACM, 2014.

[22] S. Schutte and K. Donnay. Matched wake analysis: finding causal relationships in spatiotemporal event data. *Political Geography*, 41:1–10, 2014.

[23] 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.

[24] B. Wilson, D. Cappelleri, T. W. Simpson, and M. Frecker. Efficient Pareto Frontier Exploration using Surrogate Approximations. *Optimization and Engineering*, 2(1):31–50, 2001.

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