Fining Quality in Quantity: The Challenge of Discovering Profitable 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 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 are typically heterogeneous both in their schema and the data they provide, 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, users must (1) identify sources that are relevant to their applications, (2) discover sources that collectively 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 profitable 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 profitable sources for user specified applications. We argue that the proposed system can dramatically simplify the *Discover-Appraise-Evaluate* interaction loop that many analysts 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 risen because of the ease of publishing data on the Web (e.g., search engines and online stores collect user-specific data), 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, analysts (e.g., political or business analysts) to acquire and analyze data from multiple data sources. However, a lot a valuable data is not free or open-access.

Given the sheer 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 [9, 19], they may provide duplicate data at different prices, and may exhibit significant heterogeneity in the representation of stored data, both at the schema and instance level [5, 19, 7]. The above give rise to the natural question of how can one discover *profitable sources*, i.e., sources that if integrated together the benefit for the user's application will be maximized while the corresponding cost will be minimized. Recent work [10, 24] showed how, given a fixed data domain, the benefit of integration can be specified 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 across different quality metrics. Yet 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 general 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 what are 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. [25].

EXAMPLE 1. *Assume 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 news papers, thus, our system should allow the political scientist to discover the news papers whose news articles will provide her with sufficient data either supporting or confronting her theory. The completeness, i.e., coverage, and accuracy of data are important here but the freshness of data is not 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 right trade-off between accuracy and coverage is not well known in ad-*

vance. 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 what is the individual quality and cost contribution of each selected source to the final set. 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 for the user to evaluate the solutions returned by the source management system. Finally, the previous techniques focused on fixed domains and did not support source selection for arbitrary applications in multi-user environment with diverse tasks.

In this paper we introduce our vision for a *source management system* that will enable users to specify heterogeneous integration tasks as free-text 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 provide an overview of the architecture of a data source management system and the key functionalities it should support and discuss the main challenges in building such a system, including, defining and computing multiple quality metrics that characterize the content of sources, supporting diverse integration tasks expressed as free-text queries, and enabling the interactive exploration of data sources. Then, Section 3 presents a description of our proposed data source management 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 PROFITABLE SOURCES

In this section we provide an overview of the architecture a data source management should have and the functionalities it should support and present the key challenges in each of these operations.

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 extractor module, (ii) a source analysis engine and (iii) a query engine. The basic operations of a data source management system can be separated in an *offline* and an *online* phase. During the offline phase the data extractor 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, evaluate their quality with respect to a collection of data quality metrics and keeps 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 profitable sources for her application. The aforementioned

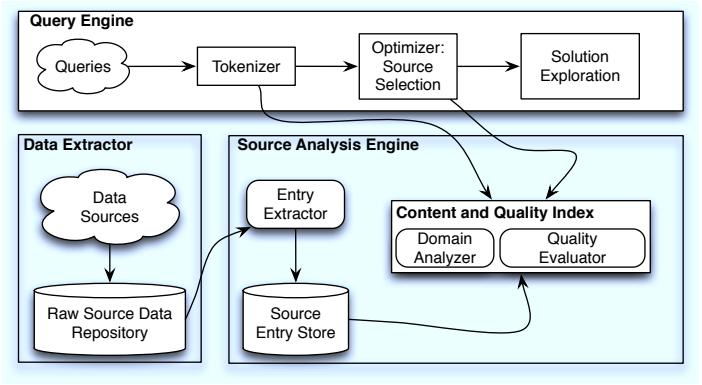

Figure 1: Source Management System Architecture.

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 profitable sources using source selection. Typically a user would start by providing the following information about her application and the desired data sources: (i) a free-text description of the task corresponding to a collection of mentions to either abstract concepts (e.g., "commerce treaties") or specific objects, 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 as input, the query engine 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 integration profit under the imposed quality and budget requirements of the user. 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 change the returned solution by removing sources or 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

Next, we 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. Being able to analyze diverse sources, both in terms of content and data type, is necessary for a multi-user environment where users with varying requirements interact with the system. For example, the data source management system 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 and for each discover the corresponding domain. 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. To do this the proposed system needs to access the raw data from each data source. Accessing and analyzing the entire content of a source is not 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 on the different concepts covered by a data source, how often should one obtain and analyze content samples to identify the rate of content change for a source and how can one determine the right samples to obtain an accurate estimate of the sources quality.

2.2.2 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 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 quality metrics to assess the quality of data [22]. 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 [10, 24].

Measuring the quality of a data source requires comparing its content with the content of the *data universe* whose data entries the source is reporting [24]. Given an integration task $\mathcal{I}$ let its data universe $\mathcal{U}(I)$ be the set of all real-world objects that are relevant to task $\mathcal{I}$. Moreover, let $\mathcal{D}(\mathcal{U}(I))$ be the domain of $\mathcal{U}(I)$ corresponding to a set of concepts (e.g., the domain can be described by the set {location, sports type}) and $\mathcal{U}(I).e$ be the set of real-world objects from $\mathcal{I}$ that correspond to an element $e \in \mathcal{D}(\mathcal{U}(I))$. 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 may correspond to a location or an organization or a time point. Moreover, we assume that either the objects in $\mathcal{U}(I)$ or their values may change over time. Given a set of sources $\bar{S}$ relevant to task I and an integration model F we have that $F(\bar{S}) \subseteq \mathcal{U}(I)$, where $F(\bar{S})$ denotes the set of objects extracted after integrating all sources in $\bar{S}$. Using this notation we define the following metrics:

Coverage. The coverage of $\bar{S}$ with respect to $\mathcal{U}(I)$ can be defined as a multinomial distribution over all elements of $\mathcal{D}(\mathcal{U}(I))$ where the probability value for an element $e \in \mathcal{D}(\mathcal{U}(I))$ corresponds to the probability that an object chosen from $\mathcal{U}(I).e$ uniformly at random will be present in $F(\bar{S})$. Notice that we do not require that the attributes of the object $r.A$ are correct.

Accuracy. The accuracy of $\bar{S}$ with respect to $\mathcal{U}(I)$ can be defined as a multinomial distribution over all elements of $\mathcal{D}(\mathcal{U}(I))$ where the probability value for an element $e \in \mathcal{D}(\mathcal{U}(I))$ corresponds to the probability that an object chosen from $F(\bar{S})$ uniformly at random is correct with respect to $\mathcal{U}(I).e$. The latter means that the object must be present in $\mathcal{U}(I).e$ and all its attributes mentioned in $\bar{S}$ should have the correct values. This definition of accuracy is equivalent to the traditional definition of accuracy focusing on erroneous values [10] and the metric of freshness (i.e., the percentage of up-to-date data in a source) focusing on state data [24].

Timeliness. The timeliness of $\bar{S}$ 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 $F(\bar{S})$ with a delay of at most t time units. This quality metric can be extended to an ensemble of distributions, one for each element $e \in \mathcal{D}(\mathcal{U}(I))$.

Publication bias. The publication bias of $\bar{S}$ 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 $\bar{S}$ by analyzing the publishing probability of $\bar{S}$ for all elements of $\mathcal{D}(\mathcal{U}(I))$. The higher the mass of this distribution is concentrated over a specific subset of entries from $\mathcal{D}(\mathcal{U}(I))$ the higher the bias of the source should be. The amount of bias in a source can be quantified using the entropy of the source publishing distribution. However, to identify the actual bias of the source the entire publishing distribution should be published.

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 [10, 24], 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 [3, 8, 23], 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.

Finally, there are multiple quality metrics and as mentioned above, it might be hard for users 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 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 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*. The user should be able to explore the different solutions on the Pareto front to identify which solution suits her task the best.

2.2.3 Diverse User Applications

The next challenge is supporting user queries over heterogeneous data domains. The different concepts or entities associated with user applications can vary significantly. Therefore, a data source management system should be able to reason about the semantic content of user tasks. 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. Techniques from keyword search applied to lists [21] or webtables [7] can be extended to suit our needs.

2.2.4 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 coverage of the integration result. Moreover, users may not know what is the feasible level of quality given their budget and what are the trade-offs between the solutions that focus on maximizing the different quality metrics in isolation. For example, selecting sources optimizing for accuracy may lead to 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, (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. ONTOLOGIES WITH DATA SOURCES

In this section we propose a preliminary design that aims to instantiate the source analysis and query engine module of the architecture proposed above and address the corresponding challenges. As demonstrated above one of the major challenges was reasoning about the content of sources focusing on diverse data domains. We propose using an *ontology* organized as a graph (e.g., Google's

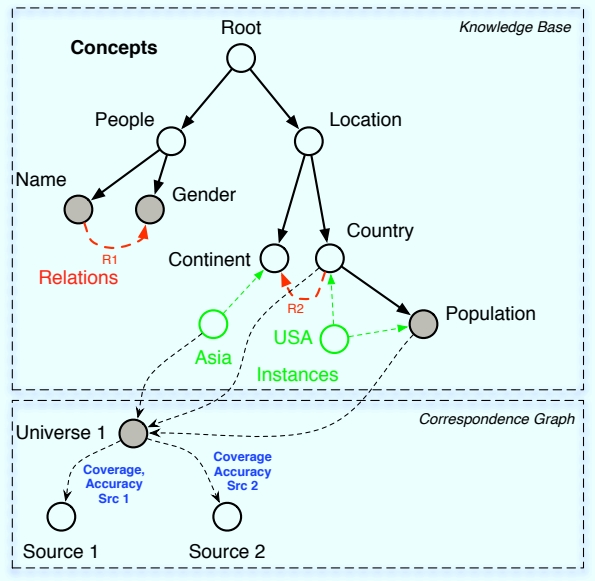

Figure 2: An example of a knowledge base and a correspondence graph with a universe node corresponding to the population of countries in Asia.

Knowledge Graph [26]) as a global relaxed schema for describing arbitrary data universes. Knowledge bases are an example of such ontology, thus, 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* 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 data universe (e.g., "Sports in the USA") can be described as a collection of concepts and/or entities. An example knowledge base is shown in Figure 2. Next, we describe how one can build the source analysis module and the query engine module, shown in Figure 1, around a knowledge base.

3.1 Source Analysis Module

To reason about the different data universes the available sources cover and the source quality we propose augmenting the knowledge base 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. Further-

more, 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 [1, 20]. Once the content of sources is represented as a collection of concept and instance conjunctions, one can use a *mixed membership model* [4] 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 [11, 2]. 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. [24] 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 2.2.2 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* [17], 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 [13] 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 base 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.

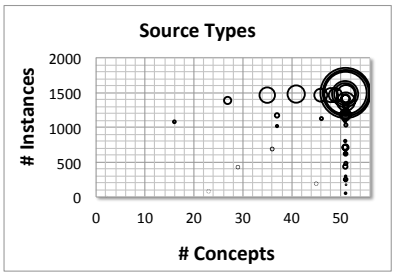

Figure 3: A fictional bubble chart describing the sources of a potential solution to a user query.

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 base. 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 universe nodes in the correspondence graph. Inferring the mixture proportions can be done using approaches similar to the ones introduce by Blei et al. [4]. 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 [10, 24]. 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 profit of integration with respect to any budget constraints. As discussed 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 [28]. Moreover, algorithms for computing the Pareto frontier of a finite set of alternatives have been studied in the skyline query literature [14, 18]. Next, we propose a two-level visualization approach for exploring solutions on the Pareto front.

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 [6, 27, 16, 15]. 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 [6, 20, 7, 29, 12] 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 [10, 24] has considered 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 numbers 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 will also enable 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] Dbpedia spotlight. https://github.com/dbpedia-spotlight/.

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

[3] 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.

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

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

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

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

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

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

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

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

[12] 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.

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

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

[15] 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.

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

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

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

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

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

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

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

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

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

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

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

[27] 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.

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

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