SPROUT^2: A Squared Query Engine for Uncertain Web Data

Robert Fink[1] and Andrew Hogue[2] and Dan Olteanu[1] and Swaroop Rath[1]

[1] Computing Laboratory, University of Oxford, Oxford, OX1 3QD, UK
[2] Google Inc., New York, NY, USA
{robert.fink,dan.olteanu,swaroop.rath}@comlab.ox.ac.uk, ahogue@google.com

ABSTRACT

SPROUT^2 is a query answering system that allows users to ask structured queries over tables embedded in Web pages, over Google Fusion tables, and over uncertain tables that can be extracted from answers to Google Squared. At the core of this service lies SPROUT, a query engine for probabilistic databases. This demonstration allows users to compose and ask ad-hoc queries of their choice and also to take a tour through the system's capabilities along pre-arranged scenarios on, e.g., movie actors and directors, biomass facilities, or leveraging corporate databases.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Query Processing*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Design, Management

Keywords

Probabilistic databases, Web data management

1. QUERYING UNCERTAIN WEB DATA

A fundamental problem in data management is to uniformly process user queries on collections containing both structured data, such as enterprise data residing in relational databases, and unstructured data, such as Web data. Our approach is to adopt structured views over unstructured data. Such views are intrinsically imperfect or *uncertain*, since they need to accommodate data that may not agree on the name, type, number of attributes, or even attribute-value pairs, and that may originate from sources of varying degree of trustworthiness. For instance, a business listing appearing on Google Maps often originates from multiple sources, e.g. Yellow Pages, phone companies, business owners themselves, or user generated content; they complement and sometimes contradict each other. Similarly, Google Squared presents answers in tabular form to keyword queries produced by aggregating possibly contradicting data from various Web sources. When compared to offline data, uncertain Web data may have the advantage of exposing

SIGMOD'11, June 12–16, 2011, Athens, Greece.
ACM 978-1-4503-0661-4/11/06.

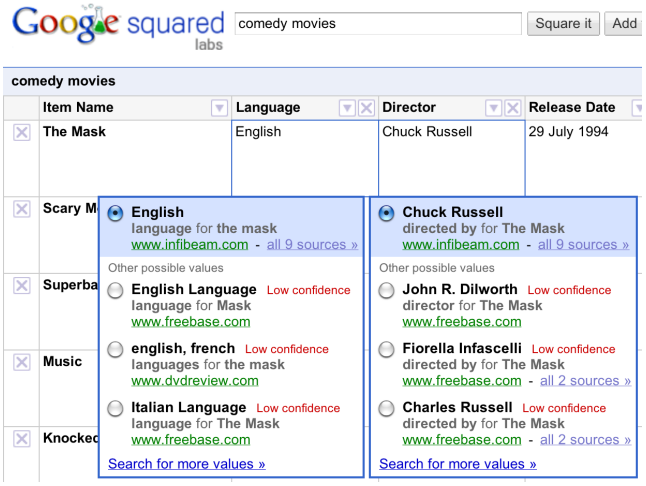

Figure 1: Snapshot of Google square *comedy movies* taken in December 2010. The pop-ups show possible values for language and director of *The Mask*.

different possibilities or opinions about a particular topic and also of being more up-to-date. Moreover, accurate data might not be readily available or known to a majority of users, and uncertain but fast answers might suffice.

We demonstrate SPROUT^2, a query answering system built on top of the query engine SPROUT [5, 7] for uncertain relational data and of existing services that host or aggregate uncertain Web data in tabular form.

An uncertain table is a relational table where fields may have several possible values ranked by confidence. Google *squares* are prime examples of uncertain tables that aggregate unstructured web data in a tabular form. For instance, the square *comedy movies* in Fig. 1 represents a list of movie titles together with their language, director, release date, description, and running time. Fields host a set of possible values together with their source and confidence score. The confidence score of a field value depends on the quality of its sources and number of its occurrences in Google searches. For instance, four possibilities are shown for the director of The Mask, each with its sources and a high/low confidence value. Fields may display no value in case no value has been found or all found values have low confidence.

We interpret a square as a representation of a discrete probability distribution over a set of deterministic tables (or *worlds*) as follows. For each square field, there are two different interpretations of the set of its possible values: The values are either *mutually exclusive* or *independent*. In the square of Fig. 1, the possible values for *Language* are inde-

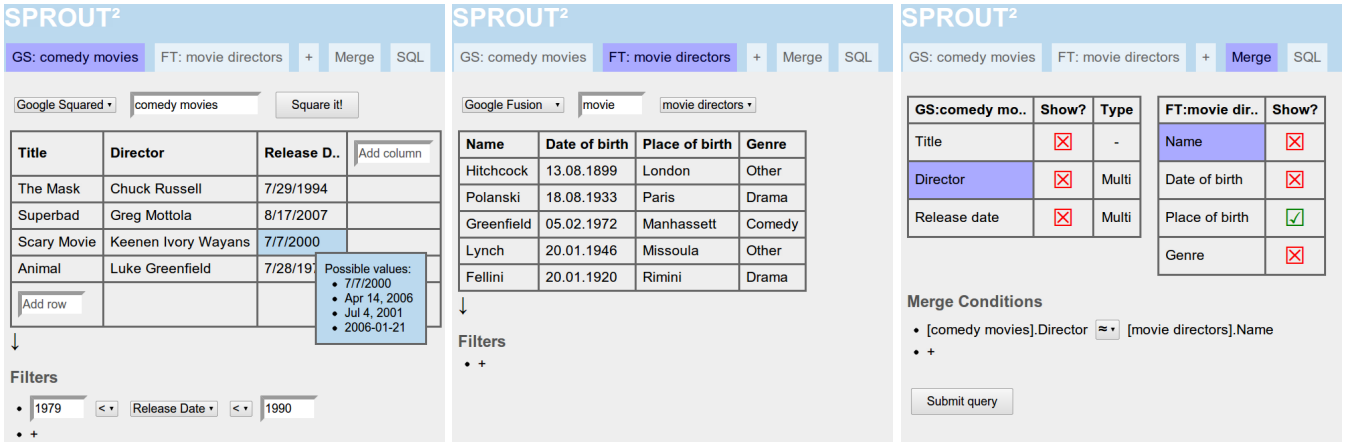

Figure 2: SPROUT² graphical user interface.

pendent, since each movie may have several languages. The possible values for *Director* of a movie are usually mutually exclusive, yet there may be movies with several directors (e.g., the Matrix). Each world is obtained by choosing for each field of the square one possible value in case of mutually exclusive values, or a subset of the possible values if they are independent. The confidences associated with field values define the probability space over the possible worlds.

SPROUT² provides a wrapper for Google Squared that can export probabilistic databases. This is useful on its own, given that a shortcoming of most existing research on uncertain data management is lack of real uncertain data.

Users can merge uncertain and deterministic data from online Google squares, Google Fusion tables, and offline data sources using relational queries with similarity joins and aggregation functions for computing confidences and expected values. For instance, one can ask for birthplaces of directors of comedy movies in the 80's. The actual query (see Fig. 1 and Section 2 for a detailed description) can be a join of the square for *comedy movies* with the Google Fusion table *movie directors* on the columns *Director* and *Name*, together with a selection on movie year and a ranking of the answers using the aggregation function for confidence computation.

2. USER INTERACTION WITH SPROUT²

SPROUT² has a visual interface that allows (i) browsing Web tables, such as those from Google Fusion Table or Google Squared, and offline tables, and (ii) visually composing queries that join such tables. Queries involving union and negation can be composed in the *SQL* tab.

Fig. 2 gives a system snapshot for the previously mentioned movies query. The user can load the Google square *comedy movies* by selecting the *Google Squared* option from the drop-down menu and entering the search term in the text field. A preview of the table including its schema is displayed. The Google Fusion table *movie directors* is added to the query by clicking the +-button in the tab bar. After entering the search phrase *movie*, a list of available tables is loaded and the user can select the table from the list. Projections and join conditions are specified in the *Merge* tab. A join can be added by clicking the +-button under merge conditions and then selecting two fields from the schema together with the join type (e.g., equality, similarity). Projections are specified by checking the respective attributes in the *Show?*-column.

Squares can be extended by both rows and columns. The users can explicitly add a new movie title or a new property of movies and Google Squared will complete the square for the corresponding rows and columns. This is especially useful when the squares do not initially have the necessary columns to express joins with other tables, or when particular movies of interest are not in the square.

The possible values of a field can be mutually exclusive or independent, cf. Sec. 1. This is not explicitly encoded in squares and SPROUT² allows the users to choose between the two semantics by switching for each column between the options *Unique* and *Multi* for mutex and independent values, respectively. Upon clicking the *Submit query* button, the query answer is computed and displayed in tabular form.

3. DEMONSTRATION SCENARIOS

SPROUT² can be useful in at least two scenarios: (1) when a user has accurate (deterministic) data available offline or from Web sources such as Google Fusion Table or AggData [1] and would like to enrich it by joining it with recent but uncertain online Web data, and (2) when all data to be joined is gathered from various Web sources in distinct squares. The advantage of using Web squares is that they present a unifying view on recent data from different, possibly contradictory sources.

Visitors to our demonstration can try ad-hoc queries on their own or from our list of worked-out scenarios. We next mention a few of them. There exists a wide range of Web sources on movie actors and directors, such as the Internet Movie Database (IMDB), Wikipedia, or user-generated content. All these sources are considered by Google Squared when answering a keyword query on movies. We could ask for instance the following queries that involve joins between squares: List most prolific directors or actors ranked by average compensation per movie, and find actors who play in an Oscar-winning movie after having played in a low-ranked movie. By joining squares with curated data collected by AggData specialists on Oscar-winning movies, we can ask for actors that won the Oscar when they were under 20.

AggData also hosts a curated table of Nobel Prize winners and their home towns. If this table is joined with appropriate squares on cities, we can answer queries listing European Nobel Prize winners, Nobel Prize winners from communities with less than 1000 inhabitants, or Nobel Prize winners that grew up in capital cities.

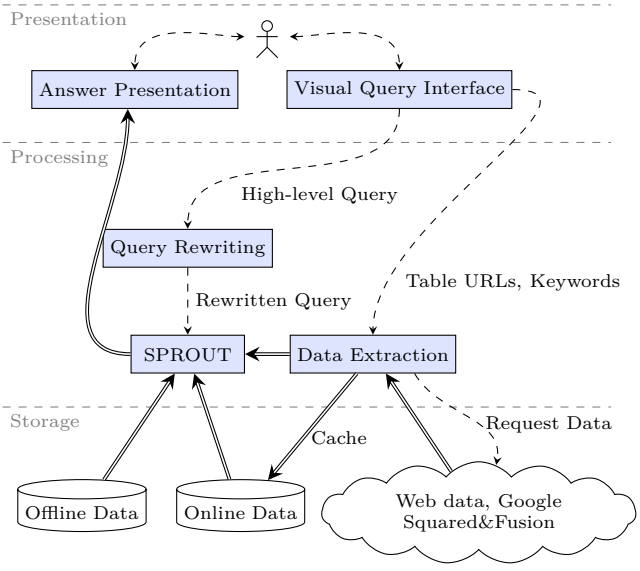

Figure 3: The SPROUT^2 architecture.

Consider a company that sells its products across Europe and has sales departments in each country. The company has an internal database with information on the number of sales employees and the revenue per country. In order to evaluate the efficiency of the sales departments in each country, a company manager can enrich the information held in the corporate database by relating her data on size and revenue of sales departments with a Google square that contains the population size per country.

A further scenario is on exploring correlations between election preferences and the density of biomass facilities in states in the USA and Canada. The Biodegradable Products Institute hosts listings of composting facilities [3]. The density of such facilities per area or per population can be obtained by joining this data with a Google square for US states that includes population size and base area. This square integrates information from various sources, such as the CIA factbook, census.gov, Wikipedia, fact-index.com, publicpurpose.com, usacitiesonline.com and others. This scenario is part of a larger set of scenarios on discovering social microtrends [6].

4. SYSTEM ARCHITECTURE

Fig. 3 depicts a conceptual view of SPROUT^2's architecture. Double and dashed arrows denote data and command flow, respectively. Designing and issuing a query in the user interface triggers (1) the rewriting of that particular query into a form compatible with the SPROUT engine and (2) the request, extraction and caching of tabular Web data as required to answer the query. The query result is then computed by SPROUT and finally presented to the user. SPROUT is a query engine for probabilistic data; it extends the backend of PostgreSQL 8.4 with exact and approximate evaluation techniques for relational algebra queries in probabilistic databases. We next describe the data extraction and query answering components.

Data extraction. This module is responsible for extracting data from given sources and compile them into uncertain tables that serve as input to the query engine or may be cached in the compiled form (*Online data* in Fig. 3). So far, we support Google squares and fusion tables, but other sources can be registered with their own wrappers. The data model of Google Squared is conceptually similar to vertically decomposed U-relations [2]. A square has an extensible schema with one distinctive key attribute (*Movie* in Fig. 1). For each other attribute, we create a table that hosts the possible values and their sources and confidences for the fields of that attribute and of the key attribute. By joining these tables on the key attribute, we can reconstitute the original square. We express the data correlations symbolically using events over random variables.

We use the rough confidence estimates proposed by Google Squared (low or high confidence) to encode the marginal probabilities of each tuple and its attributes. To further leverage Google's ranking scheme, the trustworthiness of sources of field values can be taken into account: Values from sources with high page-rank are assigned a higher marginal probability than those from low-ranked pages. This encoding strengthens the ranking scheme, since it adds correlations to the uncertain data such that tuples containing values from reliable sources are preferred.

Query answering. Queries are rewritten to account for vertical decomposition and confidence computation [2]. The rewritten queries are then evaluated using SPROUT in two steps: Firstly, the result is computed as for query evaluation in incomplete information databases: The result tuples are annotated with events created using the events of their input tuples. Here, joins create conjunctions of the input events, whereas projections and unions create disjunctions, and difference operations create negations of events. The second step is to rank the result tuples based on their probabilities. SPROUT employs novel exact and approximate knowledge-compilation techniques that compute probabilities of events for results of arbitrary relational algebra queries. The main component is an incremental compilation algorithm that repeatedly decomposes an event Φ into sub-events such that lower and upper bounds of the probability of Φ can be efficiently computed from bounds for the sub-events. The algorithm is run for a given time budget or until the desired approximation is reached [5, 4].

Acknowledgments. We thank Arnaud Sahuguet for his support and feedback on earlier drafts of this work. This research was funded by the FP7 European Research Council grant agreement FOX number FP7-ICT-233599.

5. REFERENCES

[1] AggData – FreeData Directory. Accessed Dec 2010. http://www.aggdata.com/free-data.

[2] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and Simple Relational Processing of Uncertain Data. In *ICDE*, 2008.

[3] http://www.findacomposter.com. Accessed Dec 2010.

[4] R. Fink and D. Olteanu. On the optimal approximation of queries using tractable propositional languages. In *ICDT*, 2011.

[5] R. Fink, D. Olteanu, and S. Rath. Providing support for full relational algebra in probabilistic databases. In *ICDE*, 2011.

[6] M. Penn and E. K. Zalesne. *Microtrends: The Small Forces Behind Tomorrow's Big Changes*. Twelve, 2007.

[7] The SPROUT team. *SPROUT: Scalable Query Processing in Uncertain Databases*. Oxford University, http://www.comlab.ox.ac.uk/projects/SPROUT, 2010.