Automatically Building Probabilistic Databases from the Web

Lorenzo Blanco, Mirko Bronzi, Valter Crescenzi, Paolo Merialdo, Paolo Papotti
Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione - Rome, Italy
{blanco,bronzi,crescenz,merialdo,papotti}@dia.uniroma3.it

ABSTRACT

A relevant number of web sites publish structured data about recognizable concepts (such as stock quotes, movies, restaurants, etc.). There is a great chance to create applications that rely on a huge amount of data taken from the Web. We present an automatic and domain independent system that performs all the steps required to benefit from these data: it discovers data intensive web sites containing information about an entity of interest, extracts and integrate the published data, and finally performs a probabilistic analysis to characterize the impreciseness of the data and the accuracy of the sources. The results of the processing can be used to populate a probabilistic database.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Information browsers

General Terms

Documentation, Experimentation

Keywords

Web Data Extraction, Data Integration, Probabilistic Data

1. INTRODUCTION

Existing search engines have recently started to offer services and facilities that aim at exploiting the impressive amounts of data available on the Web. For example, the results of a Google query involving a popular stock quote symbol (e.g. IBM or AAPL) are enhanced by a tuple containing the values of the major attributes (open value, day's range, volume, etc.) for that stock quote. However, enhanced answers are available only for very popular domains (finance, forecast) and for a limited number of entities (for example, only the most popular stock quote symbols are recognized). Also, since there are many sources that report attribute values for the same objects, data conflicts frequently arise. Therefore, there is the need to characterize the accuracy of the sources and the quality of the proposed data.

We have recently developed a system to exploit the huge amount of structured data available on the Web. Given a few sample pages describing instances of an entity of interest (e.g., video games, stock quotes etc.), the system is able to: (*i*) *locate* a set of web sources publishing pages with data about the entity instances; (*ii*) *extract* data from these pages; (*iii*) *integrate* the extracted data in a mediated schema; (*iv*) *analyze* the integrated data according to a probabilistic model and characterize the accuracy of the involved sources.

Our system leverages the regularities that occur in large data intensive web sites and the redundancy of data on the Web. Consider for example the pages shown in Figure 1. They come from different sources and contain structured information about video games. Sources exhibit *intra-site regularities*: they publish many pages, each containing information about one video game. Pages from the same source are generated according to a site-specific HTML template. If we abstract this representation, we may say that each web page displays a tuple, and that a collection of pages provided by the same site corresponds to a relation. According to this abstraction, each site in Figure 1 exposes its own "VideoGame" relation.

Also, observe the *inter-site information redundancy*: as the Web scales, many sources provide similar information. The redundancy occurs both a the schema level (same attributes published by several sources) and at the extensional level (several objects are published by multiple sources). In our example, at the schema level many attributes are present in almost all the sources (e.g., developer name, release date, title); while others are published by a subset of the sources (e.g., the suggested "resolution"). At the extensional level, there is a set of video games whose attribute values are published by multiple sources. As web information is inherently imprecise, redundancy also implies inconsistencies: sources can provide conflicting information for the same object (e.g., different release dates of a given video game).

Based on these observations, given an entity of interest, we may abstract the presence of a *hidden conceptual relation*, with schema $\mathcal{R}(A_1, \ldots, A_n)$, from which pages of different sources are generated: each source S_i can be seen as a view on the hidden relation.

For each source S_i, we can abstract the page generation process as the application of the following steps over the hidden relation: (*i*) selection of a subset of the tuples from the hidden relation (σ_i); (*ii*) projection on a subset of the attributes (π_i); (*iii*) introduction of errors, approximations and null values (e_i); (*iv*) encoding into web pages of the tuples of the resulting relation according to a HTML template (λ_i).

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

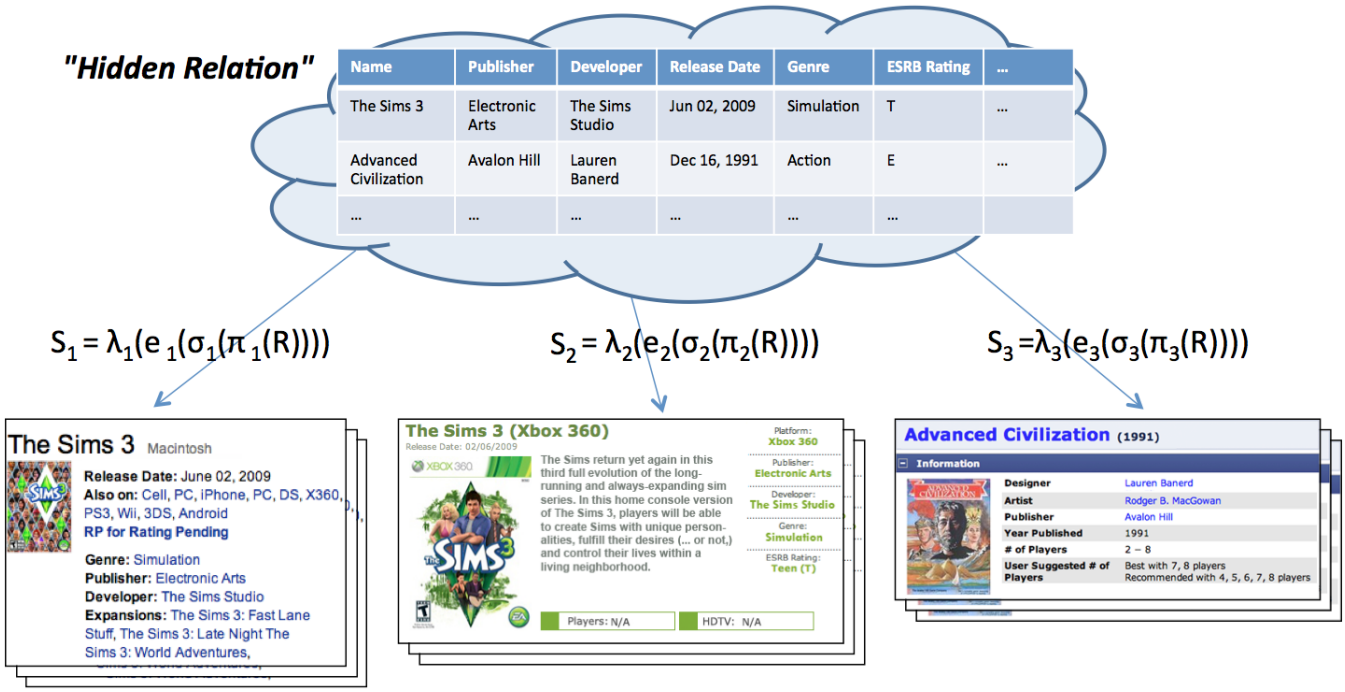

Figure 1: Given a hypothetic *hidden relation* for the entity VideoGame containing all the true values for all the attributes, we abstract the generation of three web sites (ign.com, boardgamegeek.com, teamxbox.com, respectively) as a sequence of projection, selection, approximation, and encoding over it.

From this perspective, our system addresses the following problem: *given a conceptual entity, described by means of a few input sample pages, build the underlying hidden relation.*

Our system decomposes such a challenging goal in three main steps: (*i*) *source localization*: finding all sources publishing data about the target conceptual entity and getting the pages describing its instances; (*ii*) *data extraction and integration*: inferring wrappers to extract data from each source and then integrating the redundant data; (*iii*) *probabilistic analysis*: to address the intrinsic imprecision of web data, the hidden relation is exposed in the form of a probabilistic relation: each attribute value is associated with a probability distribution function resulting from a Bayesian analysis of the conflicting values.

Our system begins with a bootstrap step to build an *initial* hidden relation $\mathcal{R}_0$ from the input sample pages. Relation $\mathcal{R}_0$ feeds an iterative process involving two steps: during the i-th iteration, the *sources localization* step uses $\mathcal{R}_i$ to locate other meaningful sources; then the *data extraction and integration* step computes $\mathcal{R}_{i+1}$ by including the new sources. As the iterations progress, it becomes harder and harder to discover new sources, and the process terminates when no more sources cannot be effectively discovered.[1]

After the last iteration, the *probabilistic analysis* step is activated by taking as input the last $\mathcal{R}_i$ produced. It returns a probabilistic relation in which a probability distribution function is assigned to each attribute of each tuple.

[1]In this context, our measure of effectiveness is simply the number of new pages that in each iteration are crawled and discarded without contributing to the hidden relation.

Related Work. To cope with the complexity and the heterogeneity of web data, state-of-the-art approaches focus on information organized according to specific patterns that frequently occur on the Web. Meaningful examples are presented in [5], which focuses on data published in HTML tables, and information extraction systems, such as TextRunner in [1], which exploits lexical-syntactic patterns. As noticed in [5], even if a small fraction of the Web is organized according to these patterns, due to the web scale the amount of data involved is impressive. Recently, the problem of truth discovery among redundant and overlapping web sources has also been faced by the Solomon system at AT&T [7]. Similarly to our probabilistic analyzer, the core of Solomon is a technique that detects copying between sources. Recently, applications that need to manage large and imprecise data sets are emerging in many different domains (e.g., sensors, RFID, information extraction). In order to store these large volumes of probabilistic data and support complex queries, probabilistic database management systems are becoming the standard solution (see [6] for a recent survey of different approaches). One of the features of our system is to automatically provide probabilistic data from the Web that are suitable for such databases.

2. SYSTEM DESCRIPTION

Figure 2 depicts the main process executed by the system. It takes as input a small set of web pages for an entity of interest, searches and processes web sources to build the hidden relation, and computes accuracy measures of the sources to assign a probability distribution to the attributes of each object. In the following we introduce the main ideas behind the techniques used to attack several issues: sources localiza-

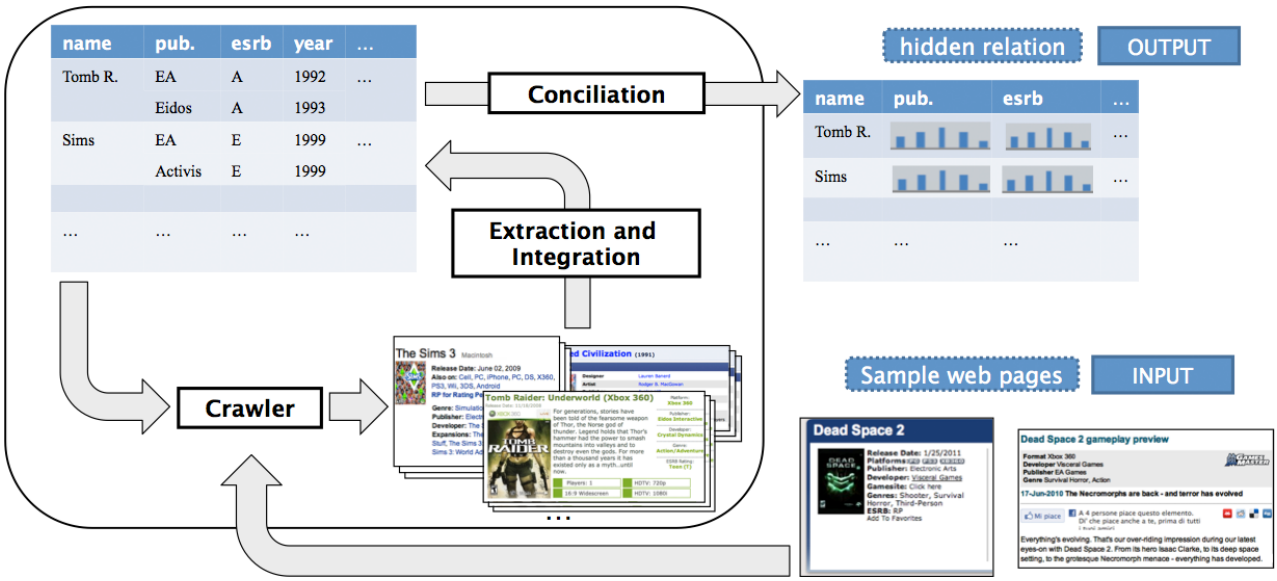

Figure 2: Overview of the system.

tion, data extraction and integration, probabilistic analysis, searching and browsing of the probabilistic repository.

Source Localization [3]. The data of the hidden relation are spread across different sources that need to be found. To achieve this goal, we developed a structure-driven crawling algorithm that locate collections of pages containing data of interest. The input of the module is the hidden relation computed by a previous iteration of the system. The crawler uses such input for two basic tasks: (*i*) querying web search engines (such as Google) to obtain new candidate sources of interest and (*ii*) analyze the search results in order to filter out the sources that are not relevant for our purposes: the system considers relevant the data intensive sources that publish information about the domain of interest. The crawler uses the data (and the metadata) in the current hidden relation to formulate the search engine queries, and infers a description of the underlying conceptual entity to filter the query results.

Extraction and Integration [2]. These challenging issues are attacked by exploiting the redundancy of data among the sources. In a bootstrapping phase, an unsupervised wrapper inference algorithm generates a set of extraction rules for each source. A instance-based matching algorithm compares data returned by the generated extraction rules among different sources and infers mappings among them. In the integration phase, the abundance of redundancy among web sources allows the system to acquire knowledge about the domain and triggers an evaluation of the mappings. Based on the quality of the inferred mappings, the matching process provides a feedback to the wrapper generation process, which is thus driven to refine the bootstrapping wrappers in order to correct imprecise extraction rules. Better extraction rules generate better mappings thus improving the quality of the solutions. Also, the system relies on the analysis of the templates performed during the process in order to extract from the page suitable semantic labels for the mappings. These labels are also used during the source localization to generate a description of the target conceptual entity.

Probabilistic Analysis [4]. As many sources use different formats, approximations, or even introduce errors, it is very common to find conflicting values for the same attribute of the same object. The computation of the correct value is not always possible (it may be the case that no source is reporting it on the Web), but a probabilistic analysis allows us to identify the most probable values based on the evidence accumulated in the hidden relation. The simplest way of reconciliate conflicting values is using a voting approach, which considers true the most frequent value. However, in the web context, this is a simplistic solution since (*i*) some sources are more reliable than others (thus they should weight more) and (*ii*) some sources may copy data from other sources and should not be considered at all.

The system performs an iterative Bayesian analysis that is able to evaluate three factors: the consensus among the sources, the sources' accuracy and the presence of copiers, that is, sources copying their data from other sources. As a result of this analysis, each source is characterized by an accuracy measure, and the hidden relation is summarized into a probabilistic relation in which attribute values are described by means of a probability distribution function.

Visualization. The generated probabilistic data can be directly used in one of the many recent probabilistic database management systems [6]. However, an important goal of the system is to assist the user with an interactive browsing of the repository. In order to provide such service, the repository can be queried with a keyword based interface and from the output probabilistic relation is navigable to verify its provenance (i.e., which sources present that value, what are the accuracy of such sources) and its probability distribution (i.e., which alternative values are published with lower probabilities) as shown in Figure 3.

3. DEMOSTRANTION

We show how end users can benefit from the system through three main demonstrative uses-cases: (*i*) discovering sources given a sample of web pages, (*ii*) extracting and integrating

structured data from the domain sources, and (*iii*) probabilistically modeling the information.

Our demonstration uses a local mirror of a fraction of the Web. In particular we will use hundreds of randomly chosen web sites containing sources for several domains.

At the beginning of the demonstration we will let the audience to pick a small number of sources publishing information about the same domain. This step corresponds to the input represented in Figure 2. As a first result, the system will produce a populated probabilistic database containing the integrated information for all the pages of the chosen sources. The database will be used to show different possible interactions with the audience at an increasing level of detail:

- in the simplest scenario, the system allows the users to perform keyword based queries on the repository. For instance, given the query "The Sims 3" in the videogames domain, the most probable values for the attributes of the instance will be shown. E.g.: publisher="Electronic Arts", release date="June 2, 2009", etc.
- in another scenario, the system allows the users to analyze the detail of the output of the probabilistic framework: for each object and each attribute returned by a query, it is possible to visualize the probability distributions with all the values published by the sources, the accuracy of the sources, the provenance of the values (snippet of the Web page). An example is shown in Figure 3.

Later we demonstrate how our system finds new sources of the target domain. The audience decides how many new web sites should be found in the local mirror of the Web. The system identifies and shows new valid sources to process. We show statistics about the discovered information (in terms of number of new tuples and new attributes in the hidden relation, etc.) and we will run queries to show the impact of the information just discovered.

On demand we will show technical insights of the three modules:

- **source localization module**: we show the description of the entity of interest, which is automatically inferred to find new sources, and how it is updated during the process;
- **extraction and integration module**: we show the extraction rules and the mappings automatically generated, and how the wrapper inference and the integration processes affect each other;
- **probabilistic analyzer module**: using both real and synthetic scenarios, we show how the probability distribution functions, the accuracy of the sources, and the presence of copiers are mutually dependent.

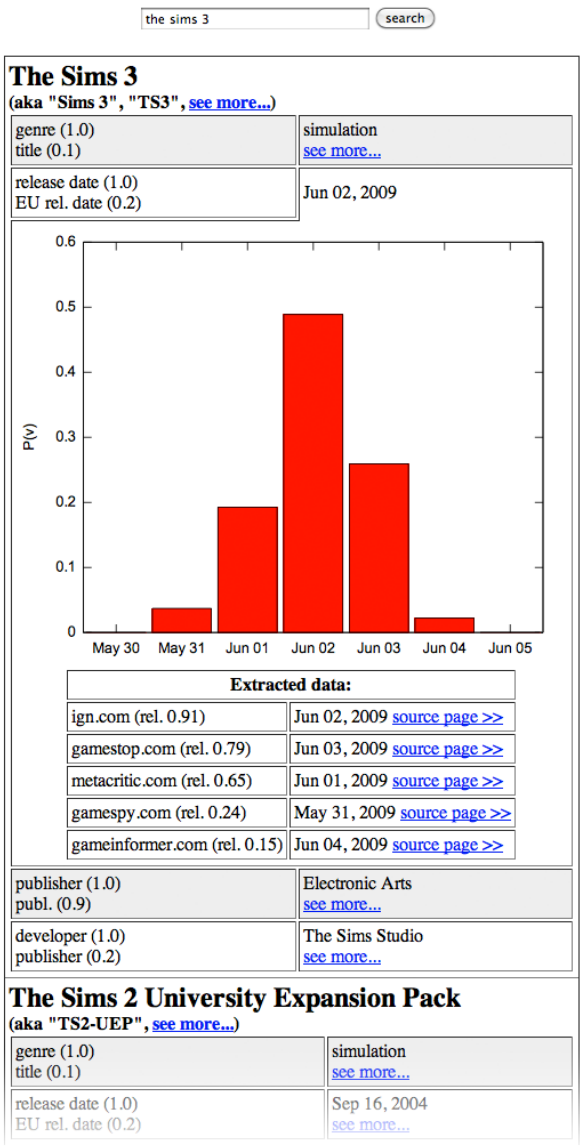

Figure 3: Screenshot of the system GUI.

4. REFERENCES

[1] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

[2] L. Blanco, M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Redundancy-driven web data extraction and integration. In *WebDB*, 2010.

[3] L. Blanco, V. Crescenzi, P. Merialdo, and P. Papotti. Supporting the automatic construction of entity aware search engines. In *WIDM*, pages 149–156, 2008.

[4] L. Blanco, V. Crescenzi, P. Merialdo, and P. Papotti. Probabilistic models to reconcile complex data from inaccurate data sources. In *CAiSE*, pages 83–97, 2010.

[5] M. J. Cafarella, A. Y. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.

[6] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: diamonds in the dirt. *Commun. ACM*, 52(7):86–94, 2009.

[7] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 3(1):1358–1369, 2010.