Exploiting Information Redundancy to Wring Out Structured Data from the Web

Lorenzo Blanco, Mirko Bronzi, Valter Crescenzi, Paolo Merialdo, Paolo Papotti
Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 — Rome, Italy
blanco, bronzi, crescenz, merialdo, papotti@dia.uniroma3.it

ABSTRACT

A large number of web sites publish pages containing structured information about recognizable concepts, but these data are only partially used by current applications. Although such information is spread across a myriad of sources, the web scale implies a relevant redundancy. We present a domain independent system that exploits the redundancy of information to automatically extract and integrate data from the Web. Our solution concentrates on sources that provide structured data about multiple instances from the same conceptual domain, e.g. financial data, product information. Our proposal is based on an original approach that exploits the mutual dependency between the data extraction and the data integration tasks. Experiments confirmed the quality and the feasibility of the approach.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Algorithms, Experimentation.

Keywords

Data extraction, data integration, wrapper generation.

1. INTRODUCTION

An increasing number of web sites deliver pages containing structured information about recognizable concepts, relevant to specific application domains, such as stock quotes, athletes, movies. Consider for example the pages shown in Figure 1, which contain information about stock quotes. As current search engines are limited in exploiting the data offered by these sources, the development of scalable techniques to extract and integrate data from fairly structured large corpora available on the Web is a challenging issue. Because of the web scale, these activities should be accomplished automatically by domain independent techniques. 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. For example, [2] focuses on data published in HTML tables, while open information extraction systems [4] exploit lexical-syntactic patterns. As noticed in [2], even if a small fraction of the Web is organized according to these patterns, because of the web scale the amount of the involved data is impressive.

We introduce an automatic, domain independent technique that exploits an unexplored publishing pattern to extract and integrate data from the Web. We concentrate on web sources that provide multiple pages about the same conceptual domain (e.g. financial data, product information, etc.) and expose data with some regularity (pages are generated from a template). Consider for example the pages reporting attributes for stock quotes (e.g., volume, last trade, etc.) in Figure 1. Each page is taken from a different site, and each site contains many other pages about stock quotes. We can abstract this representation and say that a web page displays a tuple, and that the whole collection of stock quote pages from that site corresponds to a "StockQuote" relation. Each site in Figure 1 exposes its own "StockQuote" relation as well. It is easy to experience that for many disparate real world domains the number of sites that follow this publishing strategy is huge.

Our technique to extract and integrate data from these collections of pages leverages off-the-shelf unsupervised wrapper induction algorithms (e.g. [3]), and an original instance-based data matcher to infer mappings among the data produced by the wrappers. An interesting and original feature of our approach is the exploitation of the mutual dependency between the wrapper induction and the data matching tasks: the results of the latter are used as feedback to validate and improve the extraction rules generated by the former.

Experiments on three different domains, including about 300 sites for more than 175,000 pages, demonstrate that our techniques are effective and outperform existing approaches in the quality of the final solutions.

2. OVERVIEW OF THE SOLUTION

In our framework, a source is a collection of pages generated by a common template, such that each page publishes information about one instance of a real world domain of interest. Pages in Figure 1 belong to three different sources (Yahoo!, Reuters, Google) for the stock quote domain.

A wrapper is a set of extraction rules that apply over the pages of a source: each rule extracts a string from the HTML of the page. The application of a wrapper over a page returns a tuple, and the application of a wrapper over a source returns a relation, whose schema has as many attributes as the number of extraction rules of the wrapper.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

Figure 1: Three web pages containing data about stock quotes from Yahoo!, Reuters, Google.

Given a set of sources, our goal is (*i*) to generate one wrapper for each source, and (*ii*) to correlate in *mappings* rules extracting data about the same conceptual attribute from different sources.

A natural solution to the problem is a two steps waterfall approach, where a schema matching algorithm is applied over the relations returned by automatically generated wrappers. However, important issues arise when a large number of sources is involved, and a high level of automation is required.

Wrapper Inference Problem: as wrappers are automatically generated by an unsupervised process, they can produce imprecise extraction rules (e.g., by extracting irrelevant information mixed with data of the domain).

Integration Problem: since wrappers are generated automatically, the extracted relations are "opaque", i.e., their attributes are not associated with any (reliable) semantic label. Therefore the matching algorithm must rely on an instance-based approach, which considers only attribute values to match schemas. However, in this context instance-based matching is challenging because sources provide conflicting values (due to publishing errors and heterogeneous data representation formats) and imprecise extraction rules return wrong, and thus inconsistent, data.

Our solution exploits the redundancy of data among the sources to support both the extraction and the matching steps. In a bootstrapping phase, an unsupervised wrapper inference algorithm generates a set of extraction rules for each source. A domain independent instance-based matching algorithm compares data returned by the generated extraction rules among different sources and infers mappings. The abundance of redundancy among web sources allows the system to acquire knowledge about the actual domain and triggers an evaluation of the mapping. 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 solution.

3. EXPERIMENTS

To experiment our system on real world scenarios, we collected data sources from the Web over three application domains: *Soccer Players*, *Videogames* and *Stock Quotes*. For each domain, 100 sources were gathered automatically by a crawler specifically tailored to this end [1]. Each source consists of tens to thousands of pages, and each page contains data about one instance of the corresponding domain. Within the same domain, many instances are shared by several sources. The overlap is almost total for the stock quotes because most of the sources publish all the NYSE and NASDAQ stock quotes (each stock quote appears on average in 92.8% sources), while it is more articulated for the soccer players (1.6%) and videogames (24.5%), since only popular soccer players and popular videogames are present in a large number of sources. It is worth observing that often sources provide complementary information about the overlapping instances. For example, for soccer players some sources provide weight and height, while others nationality and club.

To give a quantitative evaluation of the results, in Table 1 we report, for the 8 largest output mappings, the recall R of each mapping, i.e the number of correct extraction rules in every inferred mapping over the number of sources containing the actual attribute.

Soccer players 45,714 pages *(28,064 players)*		Videogames 68,900 pages *(25,608 videogames)*		Stock quotes 59,904 pages *(576 stock quotes)*	
Attribute	**R**	**Attribute**	**R**	**Attribute**	**R**
Name	90/100	Title	86/100	Symbol	84/99
BirthDate	61/90	Publisher	59/91	$Change	73/89
Height	54/70	Developer	45/55	%Change	73/87
Nationality	48/65	Genre	28/46	Volume	52/83
Club	43/79	EsrbRate	20/40	DayLow	43/54
Position	43/59	Rel.Date	9/31	DayHigh	41/54
Weight	34/58	Platform	9/24	LastPrice	29/50
League	14/44	#Players	6/14	OpenPrice	24/49

Table 1: Top-8 results for three domains.

For the mappings in Table 1, the system correctly assigned extraction rules to mappings with an average precision equals to 0.99, i.e. on average 99% of the rules were assigned to the correct mapping. It is interesting to report that a waterfall execution of a wrapping generation algorithm followed by an instance-based matching process produces mappings with an average precision of 0.65, 0.5 and 0.3 for the three domains of interest, respectively; while the average recall was from 6% to 11% lower than our solution.

4. REFERENCES

[1] L. Blanco, V. Crescenzi, P. Merialdo, and P. Papotti. Supporting the automatic construction of entity aware search engines. In *ACM WIDM 2008*.

[2] 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.

[3] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large Web sites. In *VLDB 2001*.

[4] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, 2008.