Contextual Data Extraction and Instance-Based Integration

Lorenzo Blanco Valter Crescenzi Paolo Merialdo Paolo Papotti
Dipartimento di Informatica ed Automazione
Università degli Studi Roma Tre
blanco,crescenz,merialdo,papotti@dia.uniroma3.it

ABSTRACT

We propose a formal framework for an unsupervised approach tacking at the same time two problems: the *data extraction problem*, for generating the extraction rules needed to gain data from web pages, and the *data integration problem*, to integrate the data coming from several sources. We motivate the approach by discussing its advantages with regard to the traditional "waterfall approach", in which data are wholly extracted before the integration starts without any mutual dependency between the two tasks.

In this paper, we focus on data that are exposed by structured and redundant web sources. We introduce novel polynomial algorithms to solve the stated problems and present theoretical results on the properties of the solution generated by our approach. Finally, a preliminary experimental evaluation show the applicability of our model with real-world websites.

1. INTRODUCTION

The development of scalable techniques to *extract* and *integrate* data from fairly structured large corpora available on the web is a challenging issue, because the web scale imposes the use of unsupervised and 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. Meaningful examples are presented in [6], which focuses on data published in HTML tables, and information extraction systems, such as TextRunner [1], which exploits lexical-syntactic patterns. As noticed in [6], 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: in their case, more than 154 millions tables were extracted from 1.1% of the considered pages.

In large data-intensive websites, we observe two important characteristics that suggest new opportunities for the automatic extraction and integration of web data:

- *local regularities*: in these sites, large amounts of data are usually offered by thousands of pages, each encoding one tuple in a local HTML template. For example, each page shown in Figure 1 comes from a different source and publishes information about a single company stock.
- *global information redundancy*: at the web scale many sources provide similar information. The redundancy occurs both a the schema level (the same attributes are published by more than one source) and at the instance level (some objects are published by more than one source). In our example, many attributes are present in all the sources (e.g., the company name, last trade price, volume); while others are published by a subset of the sources (e.g., the "Beta" indicator). At the extensional level, there is a set of stock quotes that are published by more sources. As web information is inherently imprecise, redundancy also implies inconsistencies; that is, sources can provide conflicting information for the same object (e.g., a different value for the volume of a given stock).

These observations lead us to focus on pages that are published following the *one-tuple-per-page* pattern: in each structured page you can find information about a single tuple. If we abstract this representation, we may say that a collection of structurally similar pages provided by the same site corresponds to a relation. According to this abstraction, the websites for pages in Figure 1 expose their own version of the "StockQuote" relation.[1]

Starting from the crawled web pages (for instance by using the specialized crawler introduced in [3]), our goal is: (i) transform the web pages coming from each source into a relation, and (ii) integrate these relations creating a database containing the information provided by all the sources. A state-of-the-art solution to this problem is a two-steps *waterfall* approach, where a schema matching algorithm is applied over the relations returned by automatically generated wrappers. However, when a large number of sources is involved and a high level of automation is required, important issues may arise:

- *Wrapper Inference Issues*: since wrappers are automatically generated by an unsupervised process, they can produce imprecise extraction rules (e.g., rules that extract irrelevant information mixed with data of the domain). To obtain correct rules, the wrappers should be evaluated and refined manually.
- *Integration Issues*: the relations extracted by automatically generated wrappers 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 attribute values to match schemas.

[1]For the sake of simplicity, we consider only the *one-tuple-per-page* pattern. Other variations of this pattern can be easily developed for example by preprocessing the pages with tools that fragment the HTML tables into rows [7].

Figure 1: Two web pages containing data about stock quotes from Reuters and Google finance websites.

> However, due to errors introduced by the publishing process, instance-based matching is challenging because the sources may provide conflicting values. Also, imprecise extraction rules return wrong, and thus inconsistent, data.

In [2] we presented a best-effort solution to solve these issues by taking advantage of the mutual coupling between the wrapper inference and the data integration tasks. In the present paper, we investigate the foundations of the problem and propose a principled solution based on the following contributions: (i) we propose a formal setting to state the *data extraction* and the *data integration* problems for redundant and structured web sources; (ii) we formulate a set of hypothesis that capture a few natural constraints that characterize this kind of web sources; (iii) we propose novel unsupervised polynomial algorithms to solve the stated problems whenever these hypotheses hold; (iv) we present an experimental evaluation of our model with real-world websites.

In the next section we describe a generative model of the web pages to introduce our formal setting of structured and redundant sources. Section 3 contains the algorithms to solve the integration and extraction problems in the case of perfectly overlapping sources. In Section 4 we show how removing this hypothesis affects our solution, while in Section 5 we discuss preliminary experiments with a set of sources gathered from the Web. Section 6 discusses related works and concludes the paper.

2. THE GENERATIVE MODEL

We are interested in extracting and integrating all the available information about a target entity, such as the STOCKQUOTE entity of our running example. As on the Web several sources publish information about the same entity, we can imagine that there exists a hidden relation $\mathcal{T}$, which contains all the *true information* about the objects that belong to the entity, and that sources generate their pages by taking data from $\mathcal{T}$.

We call *conceptual instances* the set of tuples $\mathcal{I}$ of the relation $\mathcal{T}$. Each tuple $I \in \mathcal{T}$ represents a real-world object of the target entity of interest. For example, in the case of the STOCKQUOTE entity, the conceptual instances of $\mathcal{T}$ model the data about the Apple stock quote, the Yahoo! stock quote, and so on. $\mathcal{T}$ has a set of attributes $\mathcal{A}$ called *conceptual attributes*. In our example they represent the attributes associated with a stock quote, such as the company name, the current trade price, the volume, and so on.

Given a set of sources $\mathcal{S} = \{S_1, \ldots, S_m\}$, each source $S_i, i = 1 \ldots m$ can be seen as the result of a generative process applied over the hidden relation $\mathcal{T}$.

The attributes published by a source are called called *physical attributes*, as opposed to the conceptual attributes of $\mathcal{T}$, and we write $S(a)$ to denote that a source S publishes a physical attribute a, and $a \in A$ to state that a physical attribute a contains data from a conceptual attribute A.

A source publishes information about a subset of the conceptual instances, and different sources may publish different subsets of its conceptual attributes.

To model the presence of conflicting data that usually occur among redundant sources, we assume that sources are noisy: they may introduce errors, imprecise or null values, over the data picked from the hidden relation. As depicted in Figure 2, for each source S_i we can abstract the page generation process as the application of the following operators over the hidden relation:

- *Selection* σ_i: returns a relation containing a subset of the conceptual instances, $\sigma(\mathcal{I}) \subseteq \mathcal{I}$.
- *Projection* π_i: returns a relation containing a subset of the conceptual attributes, $\pi(\mathcal{A}) \subseteq \mathcal{A}$.
- *Error* e_i: is a function that returns a relation, such that each correct value is kept or replaced with a null value, a synthetic value, or a value similar to the correct one.
- *Encode* λ_i: is an encoding function that produces a web page for each tuple by embedding its values into a HTML template.

The set of pages published by a source S_i can be thought as a *view* over the hidden relation, obtained by applying the above operators as follows: $S_i = \lambda_i(e_i(\pi_i(\sigma_i(\mathcal{T}))))$. From this perspective, the extraction and the integration problems can be thought in terms of these operators. The extraction becomes the inversion of the λ operator. That is, obtaining for each source S_i the associated relation $V_i = e_i(\pi_i(\sigma_i(\mathcal{T})))$. The integration becomes the problem of reconstructing $\mathcal{T}$ from the views associated with the sources.

Notice that both problems are far from being trivial as the state-of-the-art automatic wrapper inference systems are not able to create perfect wrappers, and the integration task is further complicated by the presence of errors and the absence of reliable semantic labels.

In the following we discuss under which assumptions on the intensional and extensional redundancy exhibited by the sources, our approach is able to deal with a bounded amount of error.

2.1 Intensional Redundancy

We now discuss two properties of the generative model. The first property expresses that the data published by each source are *locally consistent*. That is, within the physical attributes published by the same source, there cannot be distinct attributes with the same semantics. For example, if a website states that the current value of the stock quote "YHOO" is 17.01 there cannot be another place in the same site where you can find a different value with the same semantics. Therefore, we can write:

PROPERTY 1. Local consistency*:*
$\forall a_i, a_j \in A : S(a_i) = S(a_j) \Rightarrow a_i = a_j$
(in a conceptual attribute A there cannot exist two physical attributes coming from the same source).

The second property formalizes the presence of redundancy at the intensional level. Namely, we assume that every possible pair of conceptual attributes is published at least by one source. Let

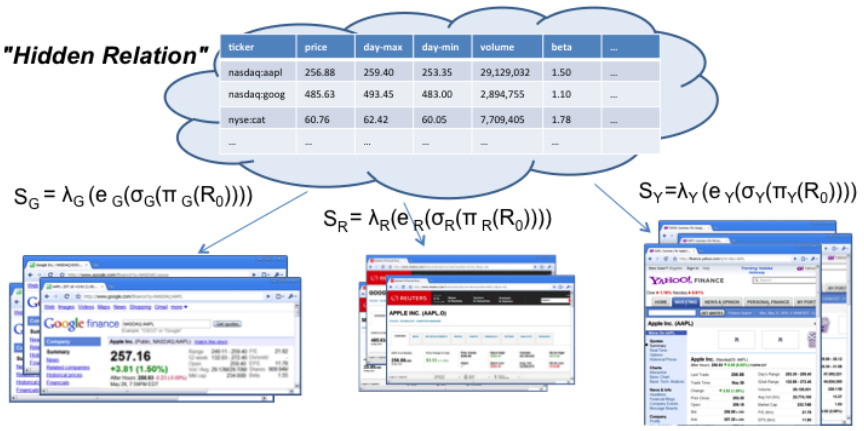

Figure 2: The publishing process: the web sources are views over the hidden relation generated by four operators.

$S(A_i)$ denotes a predicate that returns true if the source S publishes the conceptual attribute A_i. Therefore, a set of sources $\mathcal{S}$ is called *intensionally redundant* if the following property holds:

PROPERTY 2. Intensional redundancy*:*
$\forall A_i, A_j : i \neq j \; \exists \, S : S(A_i) \wedge S(A_j)$
(every possible pair of conceptual attributes is published at least by one source)

Notice that given any set of websites, this property may hold only for appropriately chosen subsets. However, the web scale represents an opportunity to gain enough redundancy: (i) in each domain, usually there is a subset of attributes that is published by most of the sources;[2] (ii) as the number of considered websites increases, the probability of meeting a new conceptual attribute decreases, and the probability of intensionally redundancy increases.

In the following we call *core*-attributes the attributes for which the intensional redundancy holds, and we call *rare*-attributes all the others.

2.2 Extensional Redundancy

At the extensional level, we assume that sources publish a common set of instances. For the sake of presentation, in the next section we rely on the hypothesis that all the source publish the same set of instances $\mathcal{I}$. Later, in Section 4 we discuss how to remove this hypothesis.

3. EXTRACTION AND INTEGRATION ALGORITHMS

Our solutions to the extraction and integration problems are discussed in this section: we propose the problem statements, the definition of solutions and the algorithms that solve them in polynomial time.

3.1 Integration Problem

We first discuss the integration problem by ignoring the extraction issues, then we discuss the main properties of the extraction rules, and, finally, how the integration and the extraction problems can be tackled together. Therefore, for the time being we assume we have a wrapper generator that is capable of perfectly inverting the encode operator λ. In other words, we do not work on web pages, but directly on the *views* of $\mathcal{T}$ published by the sources.

[2]In [11] the authors write "there is a core set of attributes that appear in a large number of items".

Given a set of sources $\mathcal{S}$, each S_i publishes a view of the hidden relation $\mathcal{T}$ such that $V_i = e_i(\pi_i(\sigma_i(\mathcal{T})))$. The integration problem can be thought as the creation of sets of physical attributes $m_1, \ldots, m_n$, called *mappings*, such that each attribute a belongs to a mapping m and each mapping contains all and only the attributes with the same semantics. The problem can be defined as follows:

PROBLEM 1. Integration Problem *: given a set of source views* $\mathcal{V} = V_1, \ldots, V_n$*, where* $V_i = e_i(\pi_i(\sigma_i(\mathcal{T})))$*, find a set of mappings* $\mathcal{M}$ *such that* $\mathcal{M} = \{m : a_1, a_2 \in m \Leftrightarrow a_1, a_2 \in A\}$.

Intuitively, we solve the problem by evaluating the physical attributes of each source and by building aggregations of attributes with the same semantics from the sources. If at the end of the process each mapping contains all and only the physical attributes with the same semantics, we have a *solution* for the problem. For example, given $a_1, a_3 \in \mathcal{V}_1$ and $a_2 \in \mathcal{V}_2$ with a_1, a_2 having the same semantics, a solution is $m_1 = \{a_1, a_2\}$ and $m_2 = \{a_3\}$.

If the sources publish correct data only, then a naive greedy algorithm easily solves the problem above. However, real sources introduce noise in the values (modeled by the error function e) that can make the integration difficult or even not possible.

To identify physical attributes with the same semantics, we rely on a *distance function* $d(a_i, a_j)$ among the values of a set of instances corresponding to the same real-world object.[3] This function compares aligned values and returns a score between 0 and 1. The more similar are the values, the lower is the distance. As the distance function works by comparing values of aligned instances, it can be easily extended to work also on conceptual attributes. We denote $d(a, A)$ the distance between the values of the physical attribute a and the values of the conceptual attribute A.

In the following, we study a class of error functions for which our algorithm can compute a solution by bounding the amount of errors that sources are allowed to introduce.

For every conceptual attribute A, let t_A denote a minimal threshold such that any physical attribute a_i belongs to A if for each $a_j \in A$, $d(a_i, a_j) < t_A$. For example, in the finance domain, a low threshold should be associated with the conceptual attribute Max value of a stock quote. This is required as there are other conceptual attributes, like the current Price and the Min value, that have similar values. On the other hand, the mapping for the trading Volume conceptual attribute can have an higher threshold since it

[3]We assume that instances can be aligned by applying a standard record-linkage technique [9].

does not usually assume values close to those of other attributes. Note that t_A is an ideal unknown threshold: it is not given as input of the integration problem and it is not necessary to know it a priori to compute the solution.

In order to solve the integration problem, it is required that the publishing errors cannot introduce enough noise to confuse the semantics of a physical attribute. We call this property as *separable semantics*:

PROPERTY 3. Separable semantics*:*
$\forall A_1, A_2, a_i \in A_1, a_j \in A_2 : a_i \neq a_j \wedge A_1 \neq A_2 \Rightarrow d(a_i, a_j) > max(t_{A_1}, t_{A_2})$
(it is possible to distinguish the semantics of physical attributes).

In order to solve the integration problem with noisy sources, we define the greedy clustering Algorithm 1.

Algorithm 1 ABSTRACTINTEGRATION

Input: A set of *locally consistent*, *intensionally redundant* sources with *separable* physical attributes.
Output: The correct set of mapping $\mathcal{M}$.

Let $G = (N, E)$ be a graph where every attribute a_i for every source $S_i \in \mathcal{S}$ is a node $n \in N$. For every pair of distinct nodes $a_i, a_j \in N$ such that $S(a_i) \neq S(a_j)$ add an edge e between them to E and let $d(a_i, a_j)$ be the weight of e.
Let $m(a_i)$ be the mapping containing the attribute a_i.

1. Add to $\mathcal{M}$ a mapping $m = \{a_i\}$ for each node $n_i \in N$,
2. insert in a list L the edges E,
3. sort L by the weight of the edges in ascending order,
4. for each edge $(a_1, a_2) \in L$:
   (a) let m be the union of the attributes in $m(a_1)$ and $m(a_2)$
   (b) if in m there is no pair of a_i, a_j such that $S(a_i) = S(a_j)$
   (c) then add m and remove $m(a_1)$, $m(a_2)$ from $\mathcal{M}$
   (d) else *break*.

We are now ready to prove the correctness of the integration algorithm.

LEMMA 3.1. ABSTRACTINTEGRATION *is correct.*

PROOF. Moved to Appendix A. □

ABSTRACTINTEGRATION is $O(n^2)$ over the total number of physical attributes, in fact most of the time is required to create the edges of the graph G.

In the following we introduce the extraction problem, that is, how to get the physical attributes we considered as input of the integration problem.

3.2 Extraction Rules

In our framework, a data source S is a collection of pages $S = p_1, \ldots, p_n$ from the same website, such that each page publishes information about one object of a real-world entity of interest.

We distinguish between two different types of values that can appear in a page: *target values*, that is, values that are derived from the hidden relation $\mathcal{T}$, and *noise values*, that is, values that are not of interest for our purpose (e.g., advertising, template, layout, etc).

We consider as given an unsupervised wrapper generator system $\mathcal{W}$. A wrapper w is an ordered set of extraction rules, $w = \{er_1, \ldots, er_k\}$, that apply over a web page: each rule extracts a string from the HTML of the page. We denote $er(p)$ the string returned by the application of the rule er over the page p. The application of a wrapper w over a page p, denoted $w(p)$, returns a tuple $t = \langle er_1(p), \ldots, er_k(p) \rangle$; therefore, the application of a wrapper over the set of pages of a source S returns a relation $w(S)$, which has as many attributes as the number of extraction rules of the wrapper. A column of the relation is a vector of values denoted $V(er_i)$: it is the result of the application of an extraction rule er_i over the pages of a source.

We say that an extraction rule er^* is *correct* if for every given page it extracts a value of the same conceptual attribute (i.e., target values with the same semantics) or a null value if the value for the attribute is missing in that page. If a correct extraction rule only extracts noise values, it is considered *noisy*. We also say that an extraction rule er^w is *weak* if it mixes either target values with different semantics or target values with noise values.

Unsupervised wrapper generators are powerful enough to infer the correct extraction rules needed to cover the data exposed by what we call *regularly structured* websites.

PROPERTY 4. Regularly structured sources*: The sources $\mathcal{S} = \{S_i\}$ are regularly structured w.r.t. a given unsupervised wrapper generator system $\mathcal{W}$, if $\mathcal{W}$ generates for each source $S_i \in \mathcal{S}$ a set of rules w_i containing all the correct rules.*

However, wrapper generators cannot automatically identify, among the generated rules, which are the correct ones. They also produce weak rules, since, at wrapper generation time there is not enough information to automatically establish if a rule is either correct or weak. The integration algorithm has been presented considering only the correct rules (i.e., physical attributes). However, noisy rules, if considered along with the correct ones, are harmless as they can be identified and deleted during the integration step. They will eventually generate singleton mappings of size one since the distance between a noisy rule and a correct rule prevent them from grouping. Similar arguments apply for distances amongst noisy rules.

Weak rules require a more detailed discussion, and unfortunately they cannot be identified and disregarded at wrapper generation step. In the following we show that, if we keep the same assumptions introduced for the integration problem, we can always identify weak rules during the integration step.

3.3 Extraction Problem

The extraction problem is defined as follows:

PROBLEM 2. Extraction Problem*: given a set of sources $\mathcal{S} = \{S_i\}$, produce a set of wrappers $W^* = \{w_i\}$, such that w_i contains all and only the correct rules for S_i.*

We now describe how we leverage the redundant information among different sources to identify and filter out the weak rules. Let er_i and er_j be two extraction rules. We say that two extraction rules "overlap" if they extract from a page the same occurrence of the same string. In this case, one of them must be a weak rule. In other terms, if many rules are extracting the same value occurrence from at least a page, only one of them is a correct rule and all the others are weak ones. With an abuse of notation, we will say that $er \in A$ to state when an extraction rule extracts at least a correct value of the conceptual attribute A. Notice that, as a weak rule er^w can extract values from n conceptual attributes, we can say $er^w \in A_1, \ldots, A_n$.

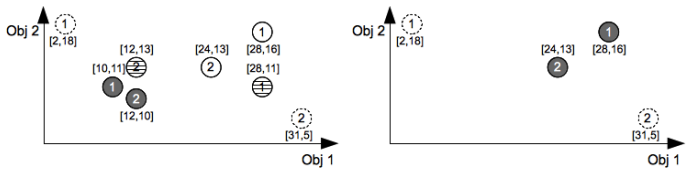

Figure 3: The Extraction algorithm in action.

These intuitions are applied in the following greedy algorithm which solves the extraction problem:

Algorithm 2 ABSTRACTEXTRACTION

Input: A set of *locally consistent*, *intensionally redundant* sources with *separable* physical attributes; the set of wrappers W produced by a wrapper generator system $\mathcal{W}$ w.r.t. which the sources are *regularly structured*.
Output: A set of wrappers W^* that do not contain weak rules.

1. **while** there is a $er \in W$ which is not marked as correct:
   (a) let $d(V(er_i), V(er_j))$ be the minimal distance between the values of two extraction rules in W and at least one of them is not marked as correct
   (b) mark er_i and er_j as correct, *(they are correct rules)*
   (c) remove from W all the rules that overlaps with er_i *(they are weak rules)*
   (d) remove from W all the rules that overlaps with er_j *(they are weak rules)*
2. now W is W^*.

The algorithm ABSTRACTEXTRACTION takes as input a set of wrappers W and computes W^* which does not contain weak rules. To explain the algorithm, we rely on the example in Figure 3. Consider two websites and two objects (say, two stock quotes) that are published by both sites. In the figure, a circle represents the values extracted by an extraction rule and its number represents the website is has been executed on. For example, in the diagram on the left, the dark circle marked with 1 extracts from the first website the values *10* for the first object and *11* for the second object. Notice that the input of the algorithm are the circles annotated with the extracted values and the website of provenience. Let say now that in step (a) the algorithm identifies the dark circles as the closest ones, and mark them as correct in step (b). At this point both the circles with horizontal stripes overlap with correct rules and can therefore be removed in steps (c) and (d). In the diagram on the right we show the resulting scenario: the dark circles are now the closest rules and are marked as correct. The remaining dashed circles do not match at all (i.e., they are noisy rules) and raise the creation of two singleton mappings. To prove that the above algorithm is correct we introduce the following lemma:

LEMMA 3.2. ABSTRACTEXTRACTION *is correct.*

PROOF. Moved to Appendix A. □

ABSTRACTEXTRACTION is $O(n^2)$ over the total number of extraction rules generated by the automatic wrapper generation system. Like in the case of ABSTRACTINTEGRATION, most of the time is spent computing distances between the extracted values.

4. NON-OVERLAPPING SOURCES

So far we have simplified the discussion by hypothesizing that every source publishes data about every object in $\mathcal{I}$. In this section we remove this simplification and use $\mathcal{I}_i^A$ to denote the subset of $\mathcal{I}$ for which S_i provides values of the conceptual attribute A.

The distances amongst physically attributes from several sources have been computed using an instance-based metric that relies on the availability of a set of shared instances between the involved sources. Therefore, if we want to compute the *direct distance* between two attributes $d(a_i, a_j)$, with $S(a_i) = S_i$ and $S(a_j) = S_j$, we need a non-empty overlap of objects between S_i and S_j ($S_i \neq S_j$), otherwise we consider $d(a_i, a_j) = \infty$.

To formalize this aspect, and given a positive integer parameter q, let $OV_{q,A}(S_i, S_j)$ be a predicate true iff $|\mathcal{I}_i^A \cap \mathcal{I}_j^A| \geq q$, i.e. both S_i and S_j publish a value of the attribute A for a shared set of at least q instances.

Intuitively, $OV_{q,A}(S_i, S_j)$ is true if we consider S_i and S_j to share enough instances (at least q) to be *directly* comparable on A's values. The value of q has to be chosen according to a trade-off: the higher the value, the more reliably the instance-based distance would perform, as it can be computed over a larger set of shared instances; however, it is possible that a lower number of sources will be directly comparable.

We are now ready to tackle the main issue: how to the compute the distance when the direct distance is not defined? i.e., the overlap is not sufficient or not available at all and $OV_{q,A}(S_i, S_j)$ does not hold.

We introduce the *indirect distance* $\overline{d}$ by leveraging the intermediate sources sharing instances with two sources not directly having enough overlap. Given a third source S_w, such that both $OV_{q,A}(S_i, S_w)$ and $OV_{q,A}(S_w, S_j)$ hold, as for the shortest path among two points is a straight line, we can easily write: $\overline{d}(a_i, a_j) \leq d(a_i, a_w) + d(a_w, a_j)$. In this case, we have an upper bound for $d(a_i, a_j)$ that we call *indirect* distance, based on the availability of two *direct* distances between (S_i, S_w) and between (S_w, S_j).

In the previous example we used just one intermediate source (S_w); the same principle can be trivially extended to a generic number of intermediate sources. However, the more intermediate sources are involved, the less precise is the bound imposed by $\overline{d}(a_i, a_j)$. In the case that we have multiple possible indirect distances, the bound chosen is the smallest one.

We call $OV_{q,A}^*$ the transitive closure of $OV_{q,A}$: $OV_{q,A}(S_i, S_j)$ is true iff it is possible to compute a distance (direct or indirect) between S_i, S_j for the attribute A. If two attributes a_i and a_j are not comparable over the same conceptual attribute A, that is, $OV_{q,A}^*(S_i, S_j)$ is false, we set $\overline{d}(a_i, a_j) = \infty$.

Let $\mathcal{S}(A)$ denote the set of websites in $\mathcal{S}$ publishing values of the conceptual attribute $A \in \mathcal{A}$: a set of websites $\mathcal{S}$ are called *extensionally redundant* if the following property holds:

PROPERTY 5. Extensional redundancy*:*
$\forall A \in \mathcal{A}\ \ OV_{q,A}^* = \mathcal{S}(A) \times \mathcal{S}(A)$
(the overlap of websites' objects allows the computation of indirect distances)

Essentially, it is required that the indirect-distance $\overline{d}$ can be computed between any pair of physical attributes.

Observe that analogously to *rare*-attributes, a concept of *rare*-instances could be introduced. Anyway, the crawler [3] used during the experiments gathers up extensionally redundant websites.

All the results previously obtained continue to hold with the new definition of distance, and can be applied even in presence of sources that do not contain all the objects $\mathcal{I}$ of the hidden relation $\mathcal{T}$ provided that the input sources are *extensional* redundant. How-

ever, in order to obtain a solution with indirect distances, we increase the complexity of the algorithms from quadratic to cubic, as we reduce the computation of the distance function to the problem of finding shortest paths in a graph by modeling physical attributes as nodes and the distances among them as weighted edges. This can be solved with the Floyd–Warshall algorithm [14].

5. EXPERIMENTAL EVALUATION

In this section we present a preliminary experimental evaluation of our model, conducted by using a special-purpose crawler [3] to collect 100 websites over three application domains: *Soccer*, *Videogames*, and *Finance*. Each source consists of tens to thousands of pages, and each page contains detailed data about one object of the corresponding entity. For each domain, we then selected the 20 largest sources and manually verified the hypothesis of our generative process.

Intensional Redundancy We start evaluating the redundancy at the schema level. We observe that in the soccer and video-game domains the majority of the conceptual attributes are not rare. In fact, in the soccer domain, only 25% of the attributes are rare and they mostly come from websites that are not only about soccer players. For example, a website containing info about olympic athletes exposes the attribute medals, while a club website exposes the debut date only for the players coming from its own youth academy. For the video-games the percentage of rare attributes is slightly over 30% of the total, in this case rare attributes come from distinct information with very similar semantics, such as difficulty and learning curve. Finally, in the stock quote scenario the percentage of rare attributes is over 40% of the total. This is not surprising, as in financial domain there is a large number of attributes (89 for 20 sources) due to the presence of many indicators used for technical analysis. For this domain, 20 sites are sufficient only to get a very rough estimation of the set of attributes published by the sites of the domain. We expect that the percentage of rare attributes would significantly drop as the number of web sources increases.

Extensional Redundancy Within the same domain, several objects are shared by several sources. The overlap is almost total for the stock quotes, while it is more articulated for soccer players and video-games as these domains include both large popular sites and small ones. Over the 100 sources, we computed that each soccer player object appears on average in 1.6 sources, each video-game in 24.5 sources, and each stock quote in 92.8 sources. In particular, $OV_{q,A}(S_i, S_j)$ is true with $q = 5$ for all the non rare attributes for all the websites.

Errors and Thresholds We manually verified the thresholds for the conceptual attributes of the three domains. As expected, those have very different values, depending on the domain and the attribute considered. As an example, in the finance domain a threshold of 0.023 is needed for the Max value of a stock quote (0.029 for the Min), while for the Volume a threshold of 0.5 is sufficient. In the soccer and video-games cases, the thresholds are higher, such as 0.44 for PlayerName (or video-game Title) and 0.36 for his BirthCountry. More importantly, we were able to verify that the *separable semantics* property is always verified.

6. RELATED WORK

Our techniques are related to projects on the integration of web data, such as PAYASYOUGO [12]. However, the proposed integration techniques are based on the availability of attribute labels, while our approach aims at integrating unlabeled data from web sites. TurboWrapper [8] has similar limits: it relies on syntactic structure of attributes (e.g. the ISBN number for books) without considering the redundancy of information that occurs at the instance-level.

The exploitation of structured web data is the primary goal of WebTables [6] and ListExtract [10], which concentrate on data published in HTML tables and lists, respectively. Compared to information extraction approaches, WebTables and ListExtract extract relations with involved relational schemas but it does not address the issue of integrating the extracted data.

Cafarella *et al.* have described a system to populate a probabilistic database with data extracted from the web [4]. However, the data are retrieved by TextRunner [1], an information extraction system that is not targeted to data rich web pages as ours. Octopus [5] and Cimple [13] support users in the creation of data sets from web data by means of a set of operators to perform search, extraction, data cleaning and integration. Although such systems have a more general application scope than ours, they involve users in the process, while our approach is completely automatic.

7. 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] M. J. Cafarella, O. Etzioni, and D. Suciu. Structured queries over web text. *IEEE Data Eng. Bull.*, 29(4):45–51, 2006.

[5] M. J. Cafarella, A. Y. Halevy, and N. Khoussainova. Data integration for the relational web. *PVLDB*, 2(1):1090–1101, 2009.

[6] 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.

[7] L. Chen, S. Ye, and X. Li. Template detection for large scale search engines. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 1094–1098, New York, NY, USA, 2006. ACM.

[8] S.-L. Chuang, K. C.-C. Chang, and C. X. Zhai. Context-aware wrapping: Synchronized data extraction. In *VLDB*, pages 699–710, 2007.

[9] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[10] H. Elmeleegy, J. Madhavan, and A. Y. Halevy. Harvesting relational tables from lists on the web. *PVLDB*, 2(1):1078–1089, 2009.

[11] J. Madhavan, S. Cohen, X. L. Dong, A. Y. Halevy, S. R. Jeffery, D. Ko, and C. Yu. Web-scale data integration: You can afford to pay as you go. In *CIDR 2007*, pages 342–350, 2007.

[12] A. D. Sarma, X. Dong, and A. Y. Halevy. Bootstrapping pay-as-you-go data integration systems. In *SIGMOD Conference*, pages 861–874, 2008.

[13] W. Shen, P. DeRose, R. McCann, A. Doan, and R. Ramakrishnan. Toward best-effort information extraction. In *SIGMOD Conference*, pages 1031–1042, 2008.

[14] S. Skiena. *The Algorithm Design Manual (2. ed.)*. Springer, 2008.

APPENDIX

A. PROOFS

We start with a preliminary lemma needed to prove the correctness of the presented algorithms.

LEMMA A.1. INTRACLOSERTHANINTER.
$\forall er_i^*, er_j^* \in A_1, er_k \in A_2\ d(V(er_i^*), V(er_j^*)) < d(V(er_i^*), V(er_k))$

PROOF. The extraction rule er_k can be correct or weak. We prove the lemma for the two cases:

1. er_k **is correct** (er_k^*): consider $er_i^*, er_j^* \in A_1$ and $er_k^* \in A_2$ when the property *separable semantics* holds.
   By definition: $\forall er_i^*, er_j^* \in A_1\ \exists\, t_{A_1} : d(V(er_i^*), V(er_j^*)) < t_{A_1}$
   Separable semantics: $\forall A_1, A_2, er_i^* \in A_1, er_k^* \in A_2 : i \neq k \wedge A_1 \neq A_2 \Rightarrow d(V(er_i^*), V(er_k^*)) > max(t_{A_1}, t_{A_2})$
   We can derive:
   $$d(V(er_i^*), V(er_j^*)) < t_{A_1} \leq max(t_{A_1}, t_{A_2}) <$$
   $$< d(V(er_i^*), V(er_k^*)).$$
   Therefore:
   $$d(V(er_i^*), V(er_j^*)) < d(V(er_i^*), V(er_k^*)).$$

2. er_k **is weak** (er_k^w): in the following we treat single values as singleton vectors that we denote with $V[i, \ldots, j]$ the subvector of values for V from index i (included) to index j (excluded). We first introduce a monotonicity property of the distance function. Given two vectors V_1 and V_2 with n values and a distance $d(V_1, V_2)$ between them, let V_2' be a copy of V_2. If we replace the i-th element $V_2[i]$ with a new element $V_2[i]'$ such that $d(V_1[i], V_2[i]) < d(V_1[i], V_2[i]')$ it follows that $d(V_1, V_2) < d(V_1, V_2')$.[4]
   In this second case er_k^w is a weak rule, that is, it can potentially contains values taken from A_1, A_2, or any other A. We consider the instance-aligned vectors of values $V_k' = V(er_k^w)$, $V_i' = V(er_i^*)$ and $V_j' = V(er_j^*)$ and we remove from the analysis the instances where er_k^w, er_i^*, and er_j^* extract the same value: let V_k, V_i and V_j be the vectors with the remaining values. As er_r^w cannot contain only values coming from A_1 (otherwise it would not be a weak rule, but a correct extraction rule of A_1) the length of these vectors must be greater than zero, and notice also that V_k' now does not contain any value coming from A_1 (they have been all removed).
   We show now by induction on the length of the vectors that $d(V(er_i^*), V(er_j^*)) < d(V(er_i^*), V(er_k^w))$.
   Base case: let $V_k[0]$ be the first value for V_k. We know that it is a correct value for a conceptual attribute different from A_1. Therefore, for the property we just showed in the previous case:
   $$d(V_i[0], V_j[0]) < d(V_i[0], V_k[0]).$$
   Inductive step: the inductive hypothesis is
   $$d(V_i[0, \ldots, n], V_j[0, \ldots, n]) < d(V_i[0, \ldots, n], V_k[0, \ldots, n]).$$
   We show that it is true for $n+1$ elements of the vectors. Again, for the property we just showed $d(V_i[n+1], V_j[n+1]) < d(V_i[n+1], V_k[n+1])$ holds. For the monotonicity property of the distance function, it is true that
   $$d(V_i[0, \ldots, n+1], V_j[0, \ldots, n+1]) <$$
   $$< d(V_i[0, \ldots, n+1], V_k[0, \ldots, n+1]).$$

[4]This is a natural property of the Euclidean distance.

□

LEMMA A.2. ABSTRACTINTEGRATION *is correct.*

PROOF. When the property *separable semantics* holds, the weights of the edges among attributes with different semantics are always higher than the weights of the edges among attributes with the same semantics. This implies that the edges in L are divided in two sublists. In the first sublist (lower weights) we have pairs of attributes that have the same semantics. We can therefore add to the solution all the pairs in the first sublist. In the second sublist (higher weights) we have pairs of attributes with different semantics and we need to avoid to add an edge from this sublist to the solution. The problem here is that it is not know a-priori where the second sublist starts. But we know that when the algorithm gets to the first edge of the second sublist, all and only the attributes with same semantics have been grouped in mappings.

Therefore the partial solution is correct.

We now need to show that the algorithm stops at the first edge of the second sublist. The first edge in the second sublist is an edge between two mappings m_1, m_2 with different semantics. When the property *intensional redundancy* holds, there must be a source which publishes two attributes a_i, a_j such that they are contained in m_1 and m_2, respectively. By the *local consistency* of sources there cannot be a mapping that contains a_i, a_j, and therefore the first edge of the second sublist is detected and the algorithm ends. □

LEMMA A.3. ABSTRACTEXTRACTION *is correct.*

PROOF. In any iteration of step (a) we select two correct extraction rules $er_1^*, er_2^* \in A_1$. This is equivalent to show that if we list the pairs of extraction rules in ascending order, the first pair is certainly one with correct extraction rules. Suppose, by absurd, that the first pair contains a weak rule. This contradicts the INTRACLOSERTHANINTER Lemma.

Every time two correct extraction rules er_1^* or er_2^* are chosen, all the weak rules containing at least a value in common with er_1^* or er_2^* are removed (steps (c) and (d)). Therefore, after the algorithm has chosen all the correct rules, there cannot be a weak rule in W as weak rules mix values shared with correct rules and they have been discarded as soon as the correct rules have been identified. □