Extraction and Integration of web sources with Humans and Domain Knowledge

Disheng Qiu, Lorenzo Luce

Dipartimento di Ingegneria
Università degli Studi Roma Tre
Via della Vasca Navale, 79 – Rome, Italy
{disheng, luce}@dia.uniroma3.it

ABSTRACT

The extraction and integration of data from many web sources in different domains is an open issue. Two promising solutions take on this challenge: *top down* approaches rely on a domain knowledge that is manually crafted by an expert to guide the process and *bottom up* approaches try to infer the schema from many web sources to make sense of the extracted data. The first solutions scale over the number of web sources, but for settings with different domains, an expert has to manually craft an ontology for each domain. The second solutions do not require a domain expert, but high quality is achieved only with a lot of human interactions both in the extraction and integration steps.

We introduce a framework that takes the best from both approaches. The framework addresses synergically both extraction and integration of data from web sources. No domain expert is required, it exploits data from a seed knowledge base to enhance the automatic extraction and integration (*top down*). Human workers from crowdsourcing platforms are engaged to improve the quality and the coverage of the extracted data. The framework adopts techniques to automatically extract both the schema and the data from multiple web sources (*bottom up*). The extracted information is then used to bootstrap the seed knowledge base, reducing in this way the human effort for future tasks.

1. INTRODUCTION

The web is a valuable source of information, but most of it is published in HTML pages. Web pages are not directly processable, so to automatically collect and process the data from these pages, two problems have to be addressed: the extraction of data from web pages and the integration of the extracted data in a unified schema. Often these issues are addressed with an extraction and integration pipeline: first a set of *wrappers* is generated for each web source, then the extracted data is matched and integrated in a unified schema.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2579707.

Many years of Data Extraction and Data Integration research are motivated by the need of scaling this process for multiple web sources in different domains at Web scale.

Nowadays, considering the Data Extraction problem, different techniques have been studied to generate *wrappers*. The common criteria adopted by all these techniques exploit the local regularities of script-generated web pages. In a data intensive web site, pages follow a common template, and this common template is exploited to infer the *wrapper* that extracts non template content. There are several Data Integration approaches, most of them exploit the published schema and its instances. The schemas of multiple web sources are first aligned and then the instances are linked.

Although there are many works both in Data Extraction and in Data Integration, to the best of our knowledge no solution has achieved the Web scale, i.e. extracting and integrating data from many web sources and with different domains in all the web. The main motivations for this deficiency is: completely automatic approaches are not accurate enough, often the extracted data generates many false positives, i.e. it has to be manually checked, and the integration process is error prone; furthermore, supervised approaches are hard to scale due to the human factor and its costs, making the process too expensive.

In a *top down* fashion, an attempt to solve these issues is by "guiding" automatic data extraction systems by adopting a knowledge base [8]. They can automatically discard non relevant data, and integrate the extracted data in a common ontology. However, they require a perfect ontology at the beginning of the extraction process, this ontology has to be manually crafted by a domain expert, i.e. a new ontology has to be built for each domain, making it hard to scale.

Another attempt to scale the extraction and integration of web data is by relying on partially overlapping web sources [2], web sources that push redundant information at schema level and at instance level. They adopt a "lazy" approach so that the schema of multiple web sources of the same domain is learned during the extraction process, in a *bottom up* fashion. They do not require a domain expert, but often the quality is not controllable and its drops when the overlap is not good enough.

The recent advent of crowdsourcing platforms, like Amazon Mechanical Turk can open new opportunities to solve these issues. The crowd can be adopted to reduce the costs of human intervention, for both Data Extraction and Data Integration. However, relying only on the crowd can make

the costs unacceptable, a fixed cost is required for each new web source.

Vision We envision a framework that takes the best from the previous approaches and aims to scale Data Extraction and Integration at Web scale. It does not require domain experts, it adopts knowledge bases available on the web (e.g. DBPedia, Freebase) as seed knowledge base to "guide" the extraction and integration process.

To improve the extraction, the framework exploits the "overlap" at schema level and at instance level of the knowledge base with multiple redundant web sources of the same domain. The system can automatically infer the schema from the sources and "map" the extracted data to the learned schema.

When the "overlap" is missing or the system is unsure of the extracted data, it poses queries to workers engaged from a crowdsourcing platform like Amazon Mechanical Turk. Crowdsourcing workers are asked to answer two simple kinds of questions: questions based on *Membership Queries*, i.e. True/False queries (e.g. "Is *Pulp Fiction* the movie title?"), and questions based on labeling (e.g. "What is *T. Fontana* for this movie?").

To reduce the human intervations, after each new web source the framework applies techniques to learn a new knowledge base from the inferred schema and its instances. The built ontology is then merged with the seed knowledge base, improving in this way its coverage for future automatic extraction processes.

We envision a full stack architecture that does not need expert users, that automatically processes web sources and poses questions to workers engaged from a crowdsourcing platform only when it is actually needed. High scalability is achieved by guiding the extraction and integration process only with simple interactions with the crowd; interactions are carefully selected to reduce the costs.

2. RELATED WORK & OPEN ISSUES

Extraction and integration are often addressed as separated problems, and following a pipeline. Data are first collected from many web sources and then integrated in a common schema.

First attempts to scale Data Extraction are based on the "template only" paradigm [1, 4]. The intuition behind these approaches is that, observing the regularity of the HTML templates, it is possibile to automatically separate the contents from the template. We are not aware of a real adoption of these solutions in production, the reason is that they require a manual check at the end of the extraction process to discard false positive, e.g. non relevant data, mistakes and so on. Supervised approaches are the most adopted solutions, for their higher quality, w.r.t., automatic approaches. However, the price to pay is in scalability, making Web scale unreachable.

A possible solution to scale supervised approaches is by relying on a knowledge base that replaces the human work. The knowledge base can be expressed like an ontology [8] or by defining annotators based on simple syntactical patterns [7]. These techniques automatically exploit the knowledge base to select the target values inside the web pages. To overcome the high error rate of the annotation process, they exploit the regularity of the HTML template. These approaches easily scale over the number of web sources which share the same knowledge base, but they require an expert user to manually craft many knowledge bases when dealing with multiple domains. However, the manual generation of a domain knowledge is expensive and error prone [10], i.e. it is hard to define a perfect ontologies that are going to work for all the websites, at the beginning of the extraction process.

An attempt to scale supervised approaches is by relying on the crowd. The cost of the manual generation of "wrappers" can be drastically reduced engaging non expert workers instead of expert ones. [5, 6] show that the crowd can be used to extract data from the web by relying on simple True/False questions. They minimize the number of questions needed to infer a "wrapper" using an Active Learning algorithm. To deal with the noise of workers engaged from a crowdsourcing platform [6] they adopt a solution inspired by the "True Finding" problem, estimating in runtime the error rate of the engaged workers. The costs are reduced, w.r.t., traditional supervised approaches, but it still requires a cost for each new web source and this cost is unacceptable when dealing with the Web scale.

An interesting solution to achieve high scalability in extraction and integration is by exploiting the redundancy of published information of multiple web sources [2], or engaging humans to improve the performances [3]. In [2] the authors observe that web sources that publish information about the same domain often show a redundancy at the schema level and a partial overlap at instance level. Aligning instances from different sources provides an automatic technique to address synergically both extraction and integration of data from web sources. This technique works only in presence of an overlap, and the quality of the results is strongly related to the quality of the alignment, i.e. if the alignment is erroneous or its number is too small, the results are uncertain and the quality drops. In the web, this overlap, is not always available. In [3] the approach massively involves expert humans, this increases its costs making it hard to scale.

Techniques to learn the knowledge base to improve the extraction process have been studied. In [11] they show that it is possible to define an ontology or bootstrap an existing one from multiple web sites of the same domain adopting simple heuristics. The extraction algorithm is too simple and it can be applied only to table like web pages.

3. ARCHITECTURE

We envision a framework that combines automatic data extraction and integration techniques [2, 8] with a supervised approach [5, 6] guided by the crowd.

Workers engaged from a crowdsourcing platform are kept in the loop of the extraction process to improve the quality and to replace the automatic approach when it fails.

A knowledge base of concepts and instances is adopted to reduce the human intervention. We adopt techniques [11] to bootstrap the starting knowledge base and improve its quality when the system processes multiple web sources of the same domain.

Figure 1 shows the architecture of the framework. To scale the framework to the Web scale, we highlight two main issues, that we discuss in this paper: the *Extraction and Matching* of instances collected from multiple web sources and the *Learning* of a new knowledge base. There are other issues that we should consider to provide a fullstack data extraction and integration system, e.g. the evaluation of work-

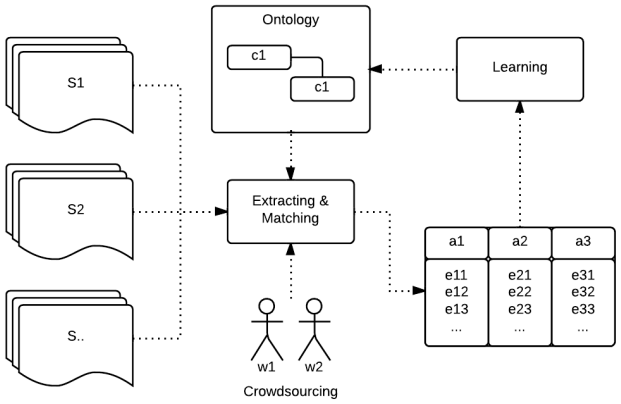


Figure 1: Framework architecture

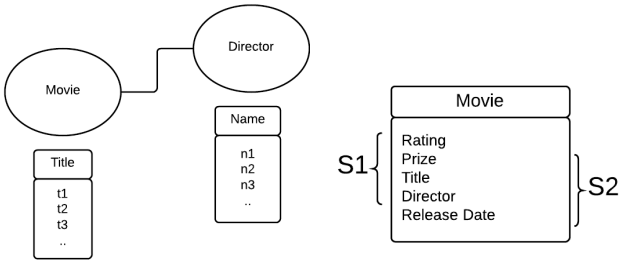


Figure 2: (Left) Seed ontology that publishes movie titles and movie directors and its instances; (Right) Attributes published by $S1$ and $S2$.

ers' error rate or the integration of conflicting data sources, but there are many works in literature that address these issues and in this paper we will discuss only the core features of our solution.

Figure 2 shows an example of a seed knowledge base and attributes published by two web sources. Web sources can publish information not present in the seed knowledge base.

3.1 Extracting and Matching

In this step, the framework extracts data from multiple web sources and matches the extracted content that refer to the same attribute. In our representation a "wrapper" is expressed as a set XPath expressions. Figure 3 shows a set of extraction rules generated by some sample pages. The system has to infer, among the generated rules, the correct rule for a target attribute. This rule is the solution for a single web source, but to integrate the data, it has also to match the rule with the rules that extract the same attribute in other web sources.

Given a web source the framework generates automatically a pool of XPath expressions for each non template textual leaf. Different techniques can be adopted to reduce the number of generated rules and to discard rules that extract non relevant template nodes [2].

Redundancy: We exploit the redundancy of the published attributes both from the web sources and the domain knowledge. In [2] the authors successfully apply their approach to align extracted values for multiple web sources. A weakness of this approach is that it extracts and matches content only if web sources publish the same attributes about the same instances. Another issue with this approach is that the aligned instances have to be representative of the variation on the HTML templates. To understand this statement consider p_1 in Figure 3, suppose that "famous" movies all follow the same template of p_1, these pages publish the attribute prize and rating not present in "non famous" movies. An alignment based on "famous" instances leads in the example to not disambiguate r_4 from r_5, and r_1 from r_2, i.e. the differences among the rules is observable only in the non famous movie, p_2.

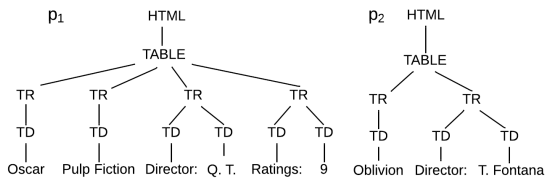
The results can suffer of a "biased" training set. In our approach we increase the probability of finding representative instances trying to align instances with an external knowledge base. In fact, instead of aligning web sources only with other web sources, we align the knowledge base with the web sources. In fact, when the overlap is bigger the quality of the learned rules can be improved.

In some settings, the overlap is still not enough, to address these issues we apply different techniques to improve its performances. In [5] the authors provide an algorithm to find representative pages inside a web source, the algorithm can be modified to find the representativeness of a set of pages, i.e. we can find if the overlap is good enough and the pages that are not represented by the overlapping training set. A possible approach to boost the performances when the representativeness is not enough, is to engage workers from the crowd. Workers are asked to answer to simple questions like "Is *Pulp Fiction* the movie title?", these questions are selected only on representative pages that are not represented by the overlap.

Human tasks: When the overlap is not representative or it is missing we rely on the crowd to extract and match correct rules. We distinguish two kind of tasks: task based on *Membership Queries* True/False questions, to infer the correct rule among a set of equivalent rules (rules that are the same for the aligned instances) and Labeling tasks, to provide a meaning to rules that are not aligned. For True/False questions, the framework can easily scale the human contribution by minimizing the number of questions needed to infer the correct rule adopting techniques based on Active Learning [5] algorithms. The algorithm selects the "best question" to ask to the crowd, so that only few tasks are required for a web source. Notice that the framework involves workers only if it is uncertain about the correctness of a rule, i.e., there are pages like p_2 that makes explicit differences among the rules.

The labeling tasks are used to make sure of the concept represented by the attribute. This is automatically done if the overlap with the knowledge base is found, but when it is missing we can exploit the crowd to provide labels for the attributes that are not aligned. We pose a question over the extracted values of non aligned rules and we ask the user to provide a label to the extracted value. In Figure 3 for p_1 it would be: "What is 9 for this movie?" and the correct label would be "rating". Obviously workers can provide different or erroneous labels, but we can apply techniques like [9] to distinguish good labels from erroneous labels engaging multiple workers.

To reduce the costs, we define techniques to reduce the number of questioned labels. A simple approach is to request a label for each extracted value, but this is too expensive. Our approach starts from the values that are extracted by more rules; we exploit the fact that rules that extract



	rule	p_1	p_2
r_1	/html/table/tr[1]/td	Oscar	Oblivion
r_2	//td[contains(.,"Director:")]/../p-s[1]:tr[2]/td	Oscar	*nil*
r_3	//td[contains(.,"Director:")]/../p-s:tr[1]/td	City of God	Oblivion
r_4	//td[contains(.,"Director:")]/../td[2]	Q.T.	T.Fontana
r_5	/html/table/tr[3]/td[2]	Q.T.	*nil*
r_6	/td[contains(.,"Rating:")]/../td[2]	9	*nil*

Figure 3: Source S1: (Left) Two sample pages from a single web source, a "famous" instance p_1 and a "non famous" instance p_2; (Right) Extraction rules, r and the values extracted by each rule in p.

the same concept often extract the same values for a subset of the pages. When the system is certain of the semantic of a group of rules, it changes the questions to True/False questions. The questions are used to select the correct rule among a group of rules.

3.2 Learning

From the *Extracting and Matching* step we collect information about the published attributes and the instances of each web source.

The goals of *Learning* step are dual: to add instances to the seed knowledge base and to add new attributes to the present concepts. Other forms of knowledge base that can be addressed are out of the scope of this paper like learning new concepts, the relationships among the concepts and axioms [11, 10]. Bootstrapping only the list of attributes and its instances improves the *Extracting and Matching* step. We believe that a minimal system can be defined only considering these two learning goals; the system can improve itself after each new web source and reduce the human intervention for future tasks.

To add new instances to the knowledge base from the extracted data, we exploit the matched instances between the knowledge base and web sources. If new attributes from the web sources are matched with the instances in the knowledge base we add the new attributes to the instances and populate the knowledge base with the extracted data.

To understand to which concept we should add the learned attributes, we follow two considerations: attributes of the same concept are often on the same HTML page, and a website that publishes information about a concept, often publishes data about the neighbor concept, and it is linked from the HTML page. We can score automatically the attributes for each concept and assign it to the concept with the best score.

Suppose that a new web source is met, the system is able to align instances from the new web source with attributes that are learned from the previous web sources. The instances in the knowledge base is now composed by instances of the initial knowledge base together with instances of the previous web sources.

4. CONCLUSIONS

We proposed a framework that synergically combines two approaches: an automatic approach that exploits the redundancy at schema level and at instance level of overlapping web sources and a supervised approach based on tasks submitted to workers engaged from a crowdsourcing platform. The crowd is adopted to improve the extraction and matching of data from web pages at the beginning of the learning process. The framework boosts the seed knowledge base by extracting data from multiple web sources, reducing in this way the dependance to human interventions for future tasks. The bootstrapped knowledge base improves the coverage of the automatic extraction and integration process for future tasks and it increases the quality of obtained results.

In this paper we have described only the core feature of the system; there are many other open issues that we have not considered in this paper, but we believe that this work can be a first step toward this direction.

5. REFERENCES

[1] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *SIGMOD Conference*, pages 337–348, 2003.

[2] M. Bronzi, V. Crescenzi, P. Merialdo, and P. Papotti. Extraction and integration of partially overlapping web sources. *Proc. VLDB Endow.*, 6(10):805–816, Aug. 2013.

[3] M. J. Cafarella, A. Halevy, and N. Khoussainova. Data integration for the relational web. *Proc. VLDB Endow.*, 2(1):1090–1101, Aug. 2009.

[4] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web sites. pages 109–118, 2001.

[5] V. Crescenzi, P. Merialdo, and D. Qiu. A framework for learning web wrappers from the crowd. In *WWW*, pages 261–272, 2013.

[6] V. Crescenzi, P. Merialdo, and D. Qiu. Wrapper generation supervised by a noisy crowd. In *DBCrowd*, pages 8–13, 2013.

[7] N. Dalvi, R. Kumar, and M. Soliman. Automatic wrappers for large scale web extraction. *Proc. VLDB Endow.*, 4(4):219–230, Jan. 2011.

[8] T. Furche, G. Gottlob, G. Grasso, O. Gunes, X. Guo, A. Kravchenko, G. Orsi, C. Schallhart, A. J. Sellers, and C. Wang. Diadem: domain-centric, intelligent, automated data extraction methodology. In *Proc. of the 21st World Wide Web Conf. (WWW)*, pages 267–270, 2012. EU Projects Track.

[9] L. von Ahn and L. Dabbish. Esp: Labeling images with a computer game. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 91–98, 2005.

[10] W. Wong, W. Liu, and M. Bennamoun. Ontology learning from text: A look back and into the future. *ACM Comput. Surv.*, 44(4):20:1–20:36, Sept. 2012.

[11] W. Wu, A. Doan, C. Yu, and W. Meng. Bootstrapping domain ontology for semantic web services from source web sites. In *Proceedings of the 6th International Conference on Technologies for E-Services*, TES'05, pages 11–22, Berlin, Heidelberg, 2006. Springer-Verlag.