WADaR: Joint Wrapper and Data Repair[*]

Stefano Ortona[#], Giorgio Orsi[#], Marcello Buoncristiano[+], and Tim Furche[#]
[#] Department of Computer Science, Oxford University, United Kingdom
[#] firstname.lastname@cs.ox.ac.uk
[+] Dipartimento di Matematica, Informatica ed Economia, Università della Basilicata, Italy
[+] marcello.buoncristiano@yahoo.it

ABSTRACT

Web scraping (or wrapping) is a popular means for acquiring data from the web. Recent advancements have made scalable wrapper-generation possible and enabled data acquisition processes involving thousands of sources. This makes wrapper analysis and maintenance both needed and challenging as no scalable tools exists that support these tasks.

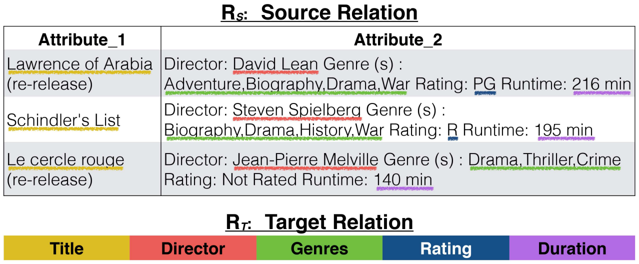
We demonstrate WADaR, a scalable and highly automated tool for joint wrapper and data repair. WADaR uses off-the-shelf entity recognisers to locate target entities in wrapper-generated data. Markov chains are used to determine structural repairs, that are then encoded into suitable repairs for both the data and corresponding wrappers.

We show that WADaR is able to increase the quality of wrapper-generated relations between 15% and 60%, and to fully repair the corresponding wrapper without any knowledge of the original website in more than 50% of the cases.

1. INTRODUCTION

Data acquisition plays an important role in modern organisations and is a strategic business process for data-driven companies such as insurers, retailers, and search engines. Nowadays, organisations acquire data in a number of ways, ranging from direct data purchase to crowdsourcing. Due to the abundance of structured data on the web, a very popular means for acquiring data is web scraping [5, 6, 8, 11], where suitable programs (called wrappers/scrapers) turn semi-structured web content into structured data. Recent advancements in the field made accurate and fully automated wrapper generation possible at scale, thus making it a good complement to direct data purchase and crowdsourcing. This revived interest has been made visible by a growing number of startups in the area, e.g., Import.io, DiffBot, as well as by recent acquisitions, e.g., BlackLocus-Home Depot, Lixto-McKinsey, and The Find-Facebook.

The large number of interesting data sources on the web, together with the progress in extraction technology call for scalable tools for maintaining the generated wrappers and the extracted data. Modern wrapper-generation systems leverage a number of features ranging from HTML and visual structures to knowledge bases and micro-data. Nevertheless, automatically-generated wrappers often suffer from errors resulting in under/over segmented data, together with missing or spurious content. As an example, Figure 1 shows the outcome of the application of a wrapper generated by RoadRunner [5] on `metacritic.com`, a review aggregator.



R_S: Source Relation

Attribute_1	Attribute_2
Lawrence of Arabia (re-release)	Director: David Lean Genre (s) : Adventure,Biography,Drama,War Rating: PG Runtime: 216 min
Schindler's List	Director: Steven Spielberg Genre (s) : Biography,Drama,History,War Rating: R Runtime: 195 min
Le cercle rouge (re-release)	Director: Jean-Pierre Melville Genre (s) : Drama,Thriller,Crime Rating: Not Rated Runtime: 140 min

R_T: Target Relation

Title	Director	Genres	Rating	Duration

Figure 1: Web data extraction with RoadRunner

The extracted (source) relation exemplifies some of the issues commonly found in wrapper-generated relations. Under and over segmentation of attributes are commonly caused by irregular HTML markups or by multiple attributes occurring within the same DOM node. Incorrect column types are instead associated with the lack of domain knowledge, supervision, or micro-data during wrapper generation. The degraded quality of the generated relations argues for means to repair both the data and the corresponding wrapper so that future wrapper executions can produce cleaner data.

We demonstrate WADaR, a system for joint wrapper and data repair for web data extraction systems. WADaR takes as input a (possibly incorrect) wrapper and a target relation schema, and iteratively repairs both the generated relations and the wrapper by observing the output of the wrapper execution. A key observation is that errors in the extracted relations are likely to be systematic as wrappers are often generated from templated websites. WADaR's repair process leverages this systematicity by iteratively: *(i)* Annotating the extracted relations with standard entity recognisers, *(ii)* Computing Markov chains describing the most likely segmentation of attribute values in the records, and *(iii)* Inducing regular expressions which re-segment the input relation according to the given target schema and that can possibly be encoded back into the wrapper.

[*]This research has received funding from the EPSRC Programme Grant EP/M025268/1 - VADA, and from the ERC PoC Grant N 641222 - ExtraLytics.

Proceedings of the VLDB Endowment, Vol. 8, No. 12
Copyright 2015 VLDB Endowment 2150-8097/15/08.

Existing approaches to joint wrapper and data repair require redundant information across multiple sources [1, 4]. WADaR replaces redundancy with an ensemble of entity recognisers which implicitly encode redundant information as they are often based on large knowledge bases or trained over large corpora. This has several advantages including the possibility to operate on each data source (and therefore each wrapper) independently, while retaining the ability to operate on different domains without retraining.

We evaluate both the performance and the generality of WADaR's approach to joint data and wrapper repair on a dataset of 100 websites from 10 different domains and by using 4 different wrapper generation systems. We show that WADaR improves the quality of the extracted relations between 15% and 60%, while being able to fully repair the corresponding wrappers in more than 50% of the cases.

Organization. Section 2 defines the problem setting and describes WADaR's repair approach. Section 3 describes the experimental setting and provides details about the accuracy and the scalability of the approach. Section 4 concludes the paper with a walkthrough of the demonstration.

2. JOINT WRAPPER AND DATA REPAIR

A wrapper W for a page P (seen as a DOM) can be represented as a set of pairs $\langle \textsc{a}, \mathcal{E} \rangle$ where $\textsc{a}$ is an *attribute* of a relation *schema* Σ_{R} ($\textsc{a} \in att(\Sigma_{\mathsf{R}})$) and $\mathcal{E}$ is an XPath expression.[1] The application of W to P, denoted by $W(P)$, produces a *relation* R. Let $\mathbf{u} = \langle u_1, \ldots, u_n \rangle \in \mathsf{R}$ be a tuple returned by $W(P)$, where u_i is the value of $\textsc{a}_i$. An *oracle* $\omega_{\textsc{a}}$ for an attribute $\textsc{a}$ is a function s.t. $\omega_{\textsc{a}}(u) = 1$ if u is in the domain of $\textsc{a}$ and $\omega_{\textsc{a}}(u) = 0$ otherwise. The *fitness* of R w.r.t. a schema Σ_{R} is defined as:

$$f(\mathsf{R}, \Sigma_{\mathsf{R}}) = \sum_{\mathbf{u} \in \mathsf{R}} \sum_{i=1}^{arity(\Sigma_{\mathsf{R}})} \omega_{\mathsf{A}_i}(u_i)$$

and counts how many times a value $u_i \in \mathbf{u}$ within a tuple is accepted by the oracles. This value is then aggregated over all tuples in the relation. The joint wrapper and data repair problem takes as inputs: a *source* relation R_s, the *wrapper* W_{R_s} that generated R_s, and a *target* schema Σ_{R_t}. An exact solution for the problem is a repaired relation R_t of maximum fitness w.r.t. Σ_{R_t}, together with a repaired wrapper W_{R_t} such that $W_{\mathsf{R}_t}(P) = \mathsf{R}_t$.

Depending on the expressiveness of the wrapper language, it may not be possible to compute an exact solution to the problem. Moreover, it can be shown that computing a relation of maximum fitness is already intractable [3, 4].

WADaR provides a general framework for joint wrapper and data repair. The approach computes an approximated solution to the problem by iterating over three steps: *annotation*, *analysis*, and *repair*.

Annotation. The annotation step identifies entities in the source relation R_s. The content of each tuple $\mathbf{u} \in \mathsf{R}_s$ (taken as a sequence of tokens) is handed over to an ensemble of entity extractors [2] replacing the oracles for the attributes in the target relation R_t. This produces an annotated relation R_s^{ω} such as the one of Figure 1.

Analysis. The analysis step takes R_s^{ω} and determines, for each position in Σ_{R_s}: *(i)* the actual sequence of attributes occurring at that position, *(ii)* the optimal position in Σ_{R_s} for extracting the values of each attribute. We then observe the following:

1. Wrappers are generated from templated web pages. As a consequence, when wrapper-generated relations have structural problems, these are usually systematic, i.e., they affect a significant part of the relation.
2. Despite being possibly inaccurate, modern entity recognisers make a limited number of non-systematic errors.

WADaR leverages the above observations by representing R_s^{ω} as a Markov chain $\mathcal{M}(S, T)$. Each state $s_{\textsc{a}} \in S$ represents an attribute type $\textsc{a} \in att(\Sigma_{\mathsf{R}})$, and the probability of a transition from $s_{\textsc{a}}$ to $s_{\textsc{b}}$ is computed as $\frac{\Pr(\textsc{b}|\textsc{a})}{\Pr(\textsc{a})}$, i.e., the probability of observing an annotation of type $\textsc{b}$ after an annotation of type $\textsc{a}$ in R_s^{ω}. This representation also allows us to tolerate non-systematic errors of the entity recognisers.

The assumption is now that highly probable sequences on the Markov chain represent hidden record structures. We iteratively compute the most-likely sequences (without repetitions) of attribute types on the Markov chain using a standard Viterbi algorithm until the induced patterns cover a sufficiently large number of tuples in R_s^{ω} (currently 85%). We then remove from R_s^{ω} all annotations that do not agree with any of the induced sequences as we assume them to be false positives of the recognisers. Both the construction of the Markov chain and the computation of the most-likely sequences can be done in polynomial time. This step updates the annotated relation R_s^{ω} in such a way that each tuple contains only one annotated value per attribute type.

It is worth noting that existing supervised approaches to, e.g., list extraction [3, 9] and data wrangling [10] can be applied to our setting to produce a repaired relation. With WADaR we wanted to explore the case where supervision is replaced by an ensemble of independently trained off-the-shelf entity extractors, thus reducing the need for large amount of training or redundant data beforehand. Another advantage is the low granularity of the entity extractors enabling portability of our approach across domains without re-training. In fact, many entity types, e.g., money, color, are often recognisable in the same way across domains.

Repair. The segmentation provided by the Markov chain, gives us enough information to compute a repaired relation R_t and the corresponding wrapper. We use the tuples of R_s^{ω} as examples for the induction of regular expressions that can be encoded in the original wrapper. For each attribute $\textsc{a} \in att(\Sigma_{\mathsf{R}_t})$, we iterate over all tuples in R_s^{ω} and compute patterns that match annotations of type $\textsc{a}$ (detailed algorithm is left for full research paper).

With reference to Figure 1, a possible expression matching values for the attribute DIRECTOR takes the content after the character ':' and before the second space, while for DURATION is one matching all characters after the last space. WADaR induces regular expressions falling within the string matching power of XPath so that they can be encoded in most of the current wrapper languages.

When a good regular expression cannot be found due to, e.g., irregularity of the examples, WADaR computes, as a fall-back strategy, an expression matching exactly the disjunction of the values provided by the entity extractors, thus simulating a dictionary for the attribute values. These expressions are called *value-based.* Value-based repairs privilege precision whereas their recall depends on the recall of the original entity recognisers.

[1] When multiple records occur on a page, e.g., on listings, a wrapper may also specify records/listing expressions.

3. EVALUATION

We evaluated WADaR to quantify the benefits of jointly repair data and wrappers.

Setting and datasets. The dataset consists of 100 websites from 10 domains and is an enhanced version of the SWDE dataset [8], a benchmark commonly used in web data extraction. SWDE's data is sourced from 80 sites and 8 domains: auto, book, camera, job, movie, NBA player, restaurant, and university. For each website, SWDE provides collections of 400 to $2k$ *detail* pages (i.e., where each page corresponds to a single record). We complemented SWDE with collections of *listing* pages (i.e., pages with multiple records) from 20 websites of real estate (RE) and used cars (UC) domains. Table 1 summarizes the characteristics of the dataset. We manually created a gold-standard (in the form of a relation) for each site, refining the ground-truth provided by SWDE where this was found to be incorrect. This can be downloaded from `http://diadem.cs.ox.ac.uk/wadar/evaluation.zip`.

Table 1: Characteristics of the dataset.

Domain	*Sites*	*Pages*	*Type*	*Records*	*Attributes*
Real Estate	10	271	Listing	3,286	15
Used Cars	10	153	listing	1,749	27
Auto	10	17,923	detail	17,923	4
Book	10	20,000	detail	20,000	5
Camera	10	5,258	detail	5,258	3
Job	10	20,000	detail	20,000	4
Movie	10	20,000	detail	20,000	4
Nba Player	10	4,405	detail	4,405	4
Restaurant	10	20,000	detail	20,000	4
University	10	16,705	detail	16,705	4
Total	100	124,715	-	129,326	78

We used four wrapper-generation systems to generate source relations, so that we can also demonstrate that WADaR is not limited to a specific wrapper language. DIADEM [6], DEPTA [11] and ViNTs [12] operate on listing pages, while RoadRunner (RR) [5] on detail pages. Despite the large amount of work in the area, we were not able to acquire any other available wrapper-generation systems.

Results. The performance is evaluated by comparing wrapper-generated relations against the gold-standard before and after the repair. The metrics used are standard Precision, Recall, and F_1-Score computed at attribute level. Figure 2 summarises the results of the evaluation. Light (resp. dark)-colored bars denote the quality of the relation, in terms of F_1-Score, before (resp. after) the repair.

From 697 attribute expressions in the input wrappers, 588 (84.4%) required some form of repair. From these, WADaR induced 335 (57.7%) correct XPath regex expressions and 253 (42.3%) value-based expressions.

A first conclusion that can be drawn from the results is that a repair is necessary in most of the cases. Before repair the quality of the data is 50% in average and never exceeds 70%. In terms of the quality of the repair, WADaR delivers a boost in F_1-Score between 15% and 60%.

In terms of system-independence, WADaR is mostly unaffected by the origin of the relations, with the only exception of ViNTs. In this case our repair strategy is hampered by two issues: *(i)* ViNTs is sometimes unable to extract some attribute values entirely, *(ii)* The values extracted by ViNTs often consist of large chunks of irregular text which affect the performance of the Markov chain computation as well as the induction of regular expressions.

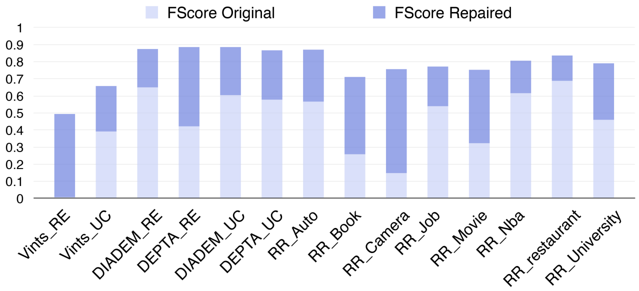


Figure 2: WADaR evaluation

Another question is whether cleanly segmented relations can be produced by direct leverage of micro-data. Only 22% of the websites in our dataset carry micro-data. In 19% of the cases micro-data is only used for site classification or to markup generic attributes such as TITLE, IMAGE, and DESCRIPTION. Only 3% of the websites markup individual attributes and would have produced a correct segmentation. In none of the cases the markups cover all attributes.

In terms of domain-independence, WADaR's performance is consistent across domains with some degradation in those domains where attribute values are multi-valued, e.g, book and camera. We plan to extend our approach in the near future to treat these cases systematically.

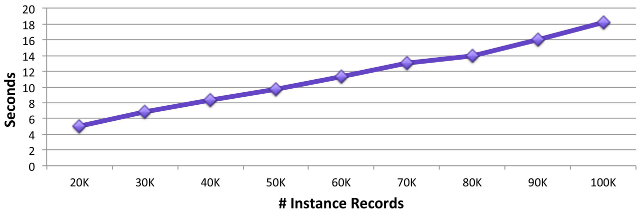


Figure 3: Running time vs input size (average).

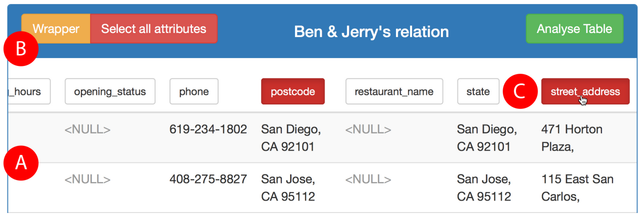
Scalability. Each step of the repair process can be parallelized at tuple level, with the only exception of the regex-induction requiring access to all tuples. Figure 3 shows a linear correlation between the repair time and the input size.

4. WADAR DEMONSTRATION

The demonstration showcases WADaR's analysis and repair tooling. WADaR supports RDBs and CSVs as well as direct execution of wrappers. WADaR's default wrapper language is OXPath [7], an extension of XPath with actions, i.e., click, fill, and extraction functions (markers).

The demonstration proceeds as follows: we first showcase examples of wrapper-generated relations from various domains, highlighting the most common errors in the data and their causes in the corresponding wrappers. As an example, one of the pre-loaded examples is a wrapper harvesting location, contact and opening hours information from the US chain `www.benjerry.com` (Figure 4). An obvious problem with this relation is that the values for LOCALITY, STATE, and POSTCODE are correct but under-segmented and duplicated (**A**). Also, STREET ADDRESS often contains spurious content. The input wrapper is accessible from the toolbar above the relation (**B**). The user can now select the attributes to be cleaned by clicking on the attribute names (**C**) or simply allow WADaR to decide which ones require cleaning by clicking on Analyse Table.

We first focus on the attributes POSTCODE and STREET ADDRESS. WADaR's analysis produces an annotated relation (Figure 5) and the user can visualize the outcome of the process for individual attributes or the whole relation (**D**). In this example, within the column POSTCODE three different entities were found: the actual POSTCODE (yellow), the LOCALITY (red), and the STATE (green). Notice that the entity recogniser for LOCALITY is incomplete as it does not recognise the value Greenbrae. Users can also access details about the Markov chain and the most likely sequences of attributes computed by WADaR (**E**).

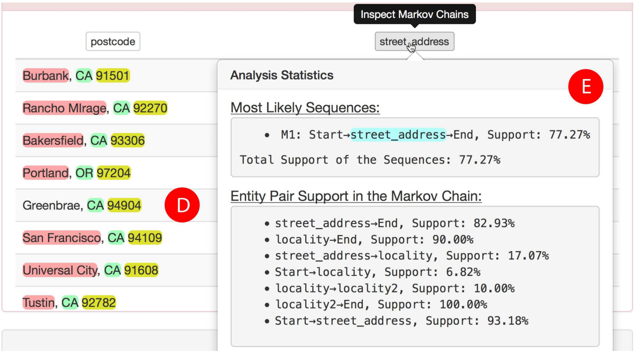


Ben & Jerry's relation

_hours	opening_status	phone	postcode	restaurant_name	state	street_address
	<NULL>	619-234-1802	San Diego, CA 92101	<NULL>	San Diego, CA 92101	471 Horton Plaza,
	<NULL>	408-275-8827	San Jose, CA 95112	<NULL>	San Jose, CA 95112	115 East San Carlos,

Figure 4: An example input relation.

The repair is then automatically computed based on the most-likely sequences and the user can inspect the proposed repair (Figure 6). WADaR highlights records and attributes where the repair was carried out successfully (**F** green) and those where a problem was detected but WADaR could not fix it (**G** red), e.g., because the computed regular expression was crossing an annotated span. If we hover the mouse over these values WADaR provides an explanation why the repair was considered unsuccessful. In this example WADaR successfully repaired LOCALITY, STATE, POSTCODE, and most of STREET ADDRESS. The user can also compare the relations before and after the repair by clicking on the Compare button above the relation (**H**) and visualise (and modify) the proposed corrections to the wrapper (**I**).



Burbank, CA 91501
Rancho MIrage, CA 92270
Bakersfield, CA 93306
Portland, OR 97204
Greenbrae, CA 94904
San Francisco, CA 94109
Universal City, CA 91608
Tustin, CA 92782

Analysis Statistics

Most Likely Sequences:

```
  • M1: Start→street_address→End, Support: 77.27%
Total Support of the Sequences: 77.27%
```

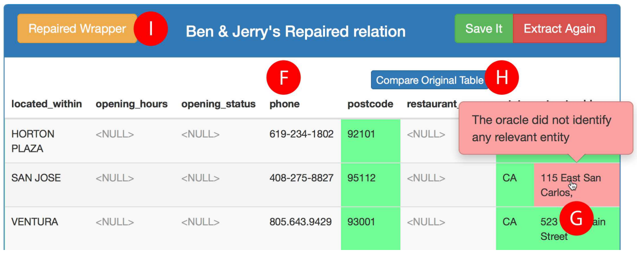
Entity Pair Support in the Markov Chain:

```
• street_address→End, Support: 82.93%
• locality→End, Support: 90.00%
• street_address→locality, Support: 17.07%
• Start→locality, Support: 6.82%
• locality→locality2, Support: 10.00%
• locality2→End, Support: 100.00%
• Start→street_address, Support: 93.18%
```

Figure 5: An annotated relation.

A separate tab reports statistical information about the repair process (Figure 7). This includes estimated error rates based on the output of the entity extractors and our confidence in their accuracy. In our example, the estimated error rate for CITY, STATE, and POSTCODE is estimated to be close to 0, while for STREET ADDRESS is 6.8%.

At this point, if the user is satisfied with the outcome of the repair, she can commit or export the repaired relation. Another possibility is to re-extract the data with the modified wrapper and repeat the process until she is satisfied with the result. The demonstration terminates with the audience picking live websites for extraction and repair.

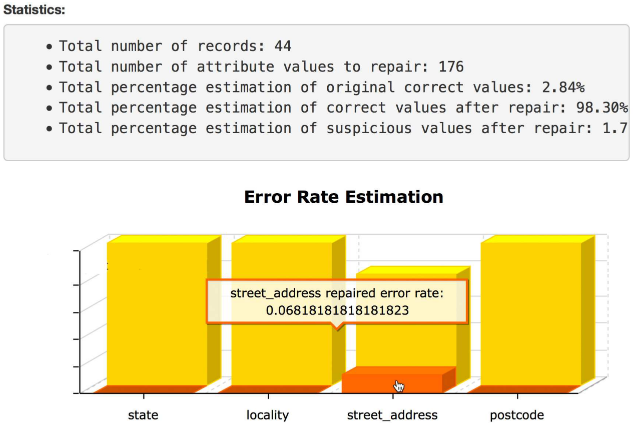


Ben & Jerry's Repaired relation

located_within	opening_hours	opening_status	phone	postcode	restaurant_		
HORTON PLAZA	<NULL>	<NULL>	619-234-1802	92101	<NULL>		
SAN JOSE	<NULL>	<NULL>	408-275-8827	95112	<NULL>	CA	115 East San Carlos,
VENTURA	<NULL>	<NULL>	805.643.9429	93001	<NULL>	CA	523 ain Street

The oracle did not identify any relevant entity

Figure 6: A repaired relation.


Statistics:

```
• Total number of records: 44
• Total number of attribute values to repair: 176
• Total percentage estimation of original correct values: 2.84%
• Total percentage estimation of correct values after repair: 98.30%
• Total percentage estimation of suspicious values after repair: 1.7
```

Figure 7: Repair statistics.

5. REFERENCES

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

[2] L. Chen, S. Ortona, G. Orsi, and M. Benedikt. Aggregating semantic annotators. *PVLDB*, 6(13):1486–1497, 2013.

[3] X. Chu, Y. He, K. Chakrabarti, and K. Ganjam. Tegra: Table extraction by global record alignment. In *SIGMOD*, pages 1713–1728. ACM, 2015.

[4] S.-L. Chuang, K. C.-C. Chang, and C. Zhai. Context-aware wrapping: synchronized data extraction. In *PVLDB*, pages 699–710, 2007.

[5] V. Crescenzi, G. Mecca, P. Merialdo, et al. Roadrunner: Towards automatic data extraction from large web sites. In *VLDB*, volume 1, 2001.

[6] T. Furche, G. Gottlob, G. Grasso, X. Guo, G. Orsi, C. Schallhart, and C. Wang. Diadem: Thousands of websites to a single database. *PVLDB*, 7(14), 2014.

[7] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, and A. J. Sellers. Oxpath: A language for scalable data extraction, automation, and crawling on the deep web. *VLDB J.*, 22(1):47–72, 2013.

[8] Q. Hao, R. Cai, Y. Pang, and L. Zhang. From one tree to a forest: a unified solution for structured web data extraction. In *SIGIR*, pages 775–784. ACM, 2011.

[9] I. R. Mansuri and S. Sarawagi. Integrating unstructured data into relational databases. In *ICDE*, pages 29–29. IEEE, 2006.

[10] D. Z. Wang, M. J. Franklin, M. Garofalakis, J. M. Hellerstein, and M. L. Wick. Hybrid in-database inference for declarative information extraction. In *SIGMOD*, pages 517–528. ACM, 2011.

[11] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *WWW*, pages 76–85, 2005.

[12] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully automatic wrapper generation for search engines. In *WWW*, pages 66–75, 2005.