

Collaborative Wrapping: A Turbo Framework for Web Data Extraction

Shui-Lung Chuang, Kevin Chen-Chuan Chang, ChengXiang Zhai

Computer Science Department

University of Illinois at Urbana-Champaign

E-mail: {schuang2, kcchang, czhai}@uiuc.edu

Abstract

To access data sources on the Web, a crucial step is wrapping, which translates query responses, rendered in textual HTML, back into their relational form. Traditionally, this problem has been addressed with syntax-based approaches for a single source. However, as online databases multiply, we often need to wrap multiple sources, in particular for domain-based integration. Observing that sources in the same domain usually share common fields, we propose a novel wrapping concept—collaborative wrapping—where multiple sources are extracted concurrently with content-based synchronization to produce consentaneous extractions. Toward this concept, recognizing wrapping as a communication process, we develop the turbo wrapper, upon the insight of turbo codes—a multi-code decoding scheme in information theory. Our experiment shows that the turbo wrapper consistently outperforms baseline single-source methods, is robust, and does benefit from extended scales of source collaboration.

1 Introduction

While the online databases multiply, much content on the Web is now provided as responses to dynamic queries, rendered in HTML pages from underlying structured data. As such data-intensive sources continue to increase, it presents an unprecedented opportunity, as well as a demand, for integration to enable effective information access.

Toward this goal, this work addresses a central barrier—wrapping across multiple sources: How to extract query results from multiple sources into *records* with segmented *fields*. Such wrapping is critical for integration, and must happen before eventually *matching* results across sources into an integrated table.

While integration naturally suggests “multiple” sources, however, current techniques (e.g., [1, 3, 4, 5]) are mostly “singular”. They build a wrapper for one source at a time, in isolation. Such *singular wrapping* must repeat wrapping

for each source, from scratch, and is thus ineffective for large-scale, domain-based scenarios.

In contrast, we argue that, since multi-source wrapping is common, where new sources may be integrated all in a *batch* or *incrementally* over time, the “multi” nature must be fundamentally supported. As the thesis of this work, we believe that, multi-source wrapping is not only necessary as a problem, but also appealing as a technique. We, therefore, propose *collaborative wrapping*, in which sources collaborate with each other to mutually correct their extraction errors and thus enhance their overall wrapping.

2 Motivation

Our insights hinge on two observations: First, *concerted convention*: As alternative sources provide related information, their query results are, not surprisingly, “similar,” by sharing common fields and values, which we refer to as a data “convention”—e.g., our survey of several domains found 64% to 75% overlapping of attributes, in average, between any two sources. Second, *complementary extraction*: From heterogeneous sources, because of their different presentations, even similar data will result in “complementary” extractions, in that their extractions (by a syntax-based wrapper, e.g., [1, 3]) tend to complement each other: A field may be hard to extract in one source but obvious in another.

Together, these observations inspire our *collaborative wrapping*. As sources are concerted and complementary, their concurrent wrapping will share “synergistic” exchange and synchronization. Using a scenario of book sources, Figure 1 contrasts the two different approaches: Given result pages as input, the traditional singular wrapping analyzes each source *in isolation*. In contrast, our approach handles all sources *collaboratively*, so that their extractions will be in consensus (e.g., Figure 1 shows cross matching between source results), as dictated by their domain convention.

3 Framework

While collaborative wrapping seems appealing, how to realize it with a principled framework? In searching for

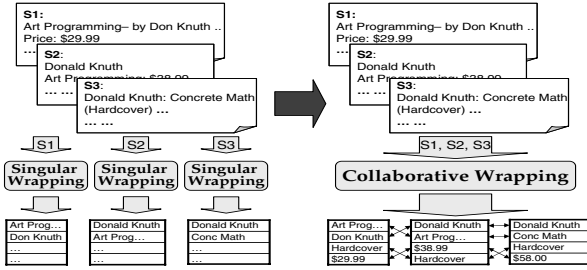


Figure 1. Two different wrapping approaches.

the insight, we realize that data wrapping can be regarded as a *communication* process: As a source renders its data in HTML pages, or *encodes* its *message* in certain *codes*, our objective is to extract its data, or *decode* the embedded *message*. This analogy enables us to learn from information theory and draw our insight from *Turbo Codes* [2], a nearly-optimal communication scheme. While the mathematics is complicated, the intuition captures the simple idea of multi-messengers: In transmitting, the *turbo encoder* will dispatch the same message through multiple error-independent “codes,” and thus, in receiving, the *turbo decoder* will recover the message by synchronizing corresponding bits between these codes.

We thus develop *turbo wrapper* to realize collaborative wrapping, paralleling the turbo-code idea. While turbo encoding *deliberately* creates multiple codes, we view our *given* sources (e.g., S_i in Figure 1) as such encoders that deliver the same message. Their extractions, much like in turbo decoding, thus must agree on and synchronize to the common message.

Figure 2 shows the framework of turbo wrapper. First, we conceptually hypothesize that there exists a domain model \mathcal{M} . For each source s , the domain model \mathcal{M} is projected to a corresponding submodel \mathcal{M}_s . Each submodel then generates a set of records, fed into *Source* s to produce pages. Then, with this conceptual model, we design our turbo wrapper to collaborate: Each *Wrapper* w , with pages as input, attempts to extract the data and decode the underlying source model, as the output, which is then utilized by other wrappers to reinforce the wrapping.

4 Experiments

We have implemented the turbo wrapper and extensively evaluated it with 30 sources in 3 domains, using 4 current singular-wrapping approaches [1, 3, 4]. The turbo wrapper constantly outperforms singular approaches, raising performance from 18-79% to 84-94% of F-measure (of precision and recall). Further, it is robust, reaching constant (over 80%) performance even starting with “weak” initial baselines. Moreover, the spirit of collaboration is indeed in ac-

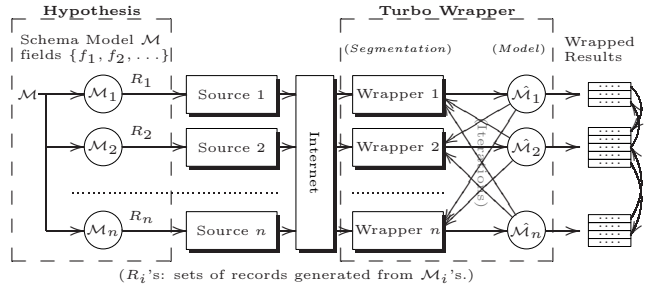


Figure 2. The framework of turbo wrapping.

tion: the more sources it handles, or the more iterations (before convergence), the better the results are.

The framework can also incorporate clean sources (e.g., previously wrapped sources) and domain knowledge (e.g., user-crafted models), thus enabling wider settings (incremental and knowledge-assisted wrapping) and further enhancing the wrapping performance (to 93–98%).

5 Conclusions

During our development of the system and experiments, we observe that the concept of our framework is general and might be useful in other applications, e.g., synchronizing and segmenting personal homepages, e.g., contact-info, or other domains, e.g., multiple gene sequence segmentation across different species. The general concept can be applied with different component instantiation for these applications.

The paper makes three contributions: (a) we propose collaborative wrapping to fundamentally support multi-source wrapping for domain-based integration. (b) To realize it, we develop an accurate wrapping approach, turbo wrapper, which enables multi-source collaborations. (c) Our empirical study shows the promise of the framework.

References

- [1] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *Proc. of SIGMOD*, pages 337–348, 2003.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. of IEEE Int. Conf. on Commun.*, pages 1064–70, Geneva, Switzerland, May 1993.
- [3] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In *Proc. of VLDB*, pages 109–118, 2001.
- [4] Y. Zhai, , and B. Liu. Web data extraction based on partial tree alignment. In *Proc. of WWW*, 2005.
- [5] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully automatic wrapper generation for search engines. In *Proc. of WWW*, 2005.