DataRover: An Automated System for Extracting Product Information From Online Catalogs

Syed Toufeeq Ahmed, Srinivas Vadrevu, and Hasan Davulcu

Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, 85287, USA
{toufeeq, svadrevu, hdavulcu}@asu.edu

Abstract. The increasing number of e-commerce Web sites on the Web introduces numerous challenges in organizing and searching the product information across multiple Web sites. This problem is further exacerbated by various presentation templates that different Web sites use in presenting their product information, and different ways of product information they store in their catalogs. This paper describes the DataRover system, which can automatically crawl and extract all products from online catalogs. DataRover is based on pattern mining algorithms and domain specific heuristics which utilize the navigational and presentation regularities to identify taxonomy, list-of-product and single-product segments within an online catalog. Next, it uses the inferred patterns to extract data from all such data segments and to automatically transform an online catalog into a database of categorized products. We also provide experimental results to demonstrate the efficacy of the DataRover.

1 Introduction

The advent of e-commerce has created a trend that has brought thousands of product catalogs online. Most of data-intensive shopping Web sites are made up of a combination of static and dynamic content, which is generated from an underlying database. Each of these data-intensive Web sites present their product information in different presentation templates with different schema. In order to effectively make use of this information, we need to organize it and make it searchable for effective mediation over the Web.

Information extraction from Web is a well-studied problem and related work can be categorized as wrapper development tools, semi-automated wrapper learning, ontology based approaches and template based automated algorithms. Wrappers [1, 2] are scripts that are created either manually or semi-automatically after analyzing the location of the data in the HTML pages. Wrappers tend to be brittle against variations and require maintenance and human intervention when the underlying Web sites change. Wrapper induction systems [3] generate extraction rules from semi-structured Web pages. These extraction rules can be
applied on other new pages to extract the data. These systems utilize the natural language processing techniques and HTML tags to infer extraction patterns. But these wrapper induction systems require labeled training examples. The template based systems employ a strong bias on the expected presentation of items within a list of products segment, such as product descriptions should reside on a single line [4] and they may not have missing or repeating attributes [4].

Many of the shopping Web sites organize their content in a taxonomy of categories and present the instances of each category in a regular fashion. A “taxonomy-directed” Web site organizes its contents into a sequence of taxonomy segments which leads to either list of items pages or single item pages. Similarly, a list of items page might lead to a set of single item pages. DataRover is based on pattern mining algorithms and domain specific heuristics which utilize these navigational and presentation regularities to identify taxonomy, list-of-products segments and single-product pages within an online catalog. Next, it uses the inferred patterns to extract data from all such segments and to automatically turn a taxonomy-directed data-intensive catalog into a database of categorized products.

For attribute extraction from single-item pages, template learning algorithms [5, 6] were developed to separate the dynamic plug in values from the static template contents. The major difference between these template learning algorithms and the DataRover’s algorithm is that: DataRover learns the path expressions of the plug in values from within the DOM trees by comparing and aligning item segments, where as RoadRunner [5] tokenizes, compares and aligns the HTML token sequences tag by tag. Our procedure of comparing segment by segment and then learning of requisite data paths requires only two examples and hence the learning is faster and also, the learned path expressions are more resilient against missing, repeated or reordered data values – where as such resilient grammar learning from token sequences requires that their learning algorithms should be presented with examples of all possible variations.