Journal of Universal Computer Science, vol. 14, no. 11 (2008), 1911-1920
submitted: 30/9/07, accepted: 25/1/08, appeared: 1/6/08 © *J.UCS*

Exploring Information Extraction Resilience

Dawn G. Gregg
(University of Colorado, Denver, CO, USA
dawn.gregg@cudenver.edu)

Abstract: There are many challenges developers face when attempting to reliably extract data from the Web. One of these challenges is the resilience of the extraction system to changes in the web pages information is being extracted from. This article compares the resilience of information extraction systems that use position based extraction with an ontology based extraction system and a system that combines position based extraction with ontology based extraction. The findings demonstrate the advantages of using a system that combines multiple extraction techniques, especially in environments where web sites change frequently and where data collection is conducted over an extended period of time.

Key Words: Information extraction, semi-structured data, ontologies
Category: H3.3, H3.4, H5.4

1 Introduction

The Internet is a source of vast amounts of information. There are company, government, organisation and individual web sites, tutorials, discussion forums, blogs, wikis, social bookmarks and social networking services. Most of these sites contain information that could potentially be useful for a variety of applications if the information could be systematically extracted and used. For example, networks of friends on social networking sites can be used by businesses to understand their customers or by governments mapping terrorist groups; data from blogs can be aggregated based on their labels and used to create a repository of knowledge on a particular topic; or social bookmarks can be used by anthropologists studying patterns in society. However, the volume of information available on the Web makes it impractical to manually process enough information for use in these types of applications. Instead users must rely on computerised tools that can systematically roam the Web and carry out sophisticated information processing tasks on their behalf.

Understanding web content is one of the major challenges developers who are interested in tapping the Internet's vast information resources must face. One approach to simplifying the process of understanding the Web is the Semantic Web. The idea behind the Semantic Web is that if web page authors add semantics to meaningful web content it will make it easier to find and use web data [Berners-Lee et al., 2001]. Adding these formal semantics to web pages will aid in everything from resource discovery to the automation of information processing tasks [Koivunen and Miller, 2002]. On the Semantic Web, systems would be able to query the Internet and obtain information of interest based on metadata written in the Resource Description Framework language,

or RDF for short [Carroll and Klyne, 2004]. However, currently, only a small fraction of web content contains the metadata necessary to make such Internet queries possible. For years people have been writing web pages in HTML creating a vast amount of human-readable content. It is unlikely that even a small fraction of these documents will ever be rewritten to include the RDF metadata necessary to allow them to be processed by Semantic Web agents [Embley, 2004]. Instead developers interested in using web data must create systems capable of using heterogeneous human-readable data.

Using information from the current human-readable Web in a systematic fashion presents numerous challenges to developers. The first challenge is locating information of interest. Typical web searches return thousands of documents related to a search term. Information retrieval systems attempt to locate interesting documents from within the thousands of uninteresting documents returned from these searches. The second challenge is extracting data from the Web in a form that allows computer systems to manipulate and derive meaning from it [Berners-Lee et al., 2001]. The third challenge is to develop extraction processes that are resilient to changes in the structure of the basic web resource. The Web is a dynamic medium and many web sites frequently change the layout of their web pages to make them more useful or attractive to the human user. Unfortunately, this can cause extraction systems designed to work with the original web site to fail. A final challenge is to develop tools that allow the information content of web sites to be understood by applications. There are a number of techniques that can be used to cluster, categorise, or otherwise derive meaning from web content [Liu and Maes, 2005, Schuff and Turetken, 2006]. However, applying these techniques is rarely straightforward and the process of developing systems to understand the meaning of human generated content is a continuously evolving science.

The focus of this article is on two of the four challenges discussed above: the extraction of data from web pages of interest and the resilience of the extraction system. This article discusses how an adaptive web information extraction system can be applied to automate information extraction in a domain that changes frequently. The Amorphic information extraction system used for this study can locate data of interest based on domain-knowledge or page structure, and can detect when the structure of a web based resource has changed and act on this knowledge to search the updated resource to locate the desired information [Gregg and Walczak, 2006a, Gregg and Walczak, 2007]. An experiment was conducted in which Amorphic is used to extract information from a web site using position based extraction, ontology based extraction and a combination of the two. The experiment demonstrates that although ontology extraction is inherently more resilient than position extraction, a combined approach provides better performance than either technique used in isolation. The remainder of the article is structured as follows: Section 2 describes related work on information extraction wrappers and wrapper repair; Section 3 describes the Amorphic information extraction system; Section 4 presents the experiment comparing the different extraction approaches and section 5 presents the discussion and conclusions.

2 Related Work

Information extraction automates the translation of input pages or text into structured data. Information extraction systems usually rely on extraction rules tailored to a particular information source. These extraction rules are called wrappers, where a wrapper is defined as a program or a rule that understands information provided by a specific source and translates it into a regular form, e.g., XML or relational tables. The wrapper allows the information extraction system to recognise the data of interest amongst many other uninteresting pieces of text [Laender et al., 2002], e.g., markup tags, in-line code, or navigation hints. Generally, information extraction can be evaluated on three different characteristics, the type of extraction document/target, the extraction technique, and the degree of automation [Chang et al., 2006]. For example, the input documents used for information extraction can be unstructured free-text documents written in natural language or semi-structured documents. Extraction can use position based wrappers or wrappers that are ontology driven. Finally, extraction systems vary in the degree of automation involved in generating the initial wrapper or in repairing the wrapper when the underlying data source changes.

2.1 Extraction Targets

There are a variety of document types from which data can be extracted, including free-form text documents like news articles, semi-structured documents like medical records or computer logs, and structured documents based on XML or RDF [Chang et al., 2006, Turmo et al., 2006]. The focus of this article is on information extraction from HTML documents available via the Web. A substantial proportion of these HTML documents are semi-structured documents either because they are dynamic pages generated from a database, e.g., an eBay auction page or a search result page, or because the human-generated content conforms to a regular pattern, e.g., manually generated blogs or lists of publications.

Semi-structured HTML data can also vary based on the way extraction targets (data of interest within the web page) are defined. For example, a web page can contain data for a single data entity or record, e.g., a single product page, or the web page can contain data for multiple data entities or records, e.g., a search results page or an organisation membership list. Web pages containing data for a single data entity use page level wrappers to extract all of the extraction targets embedded in that page. The data can be labelled or unlabelled, exist in tables, lists, or other formats. Web pages containing multiple records use record level wrappers to discover record boundaries then divide each record into individual attributes [Embley et al., 1999, Sarawagi, 2002]. Multiple-record data can be in regularly formatted tables with column headings, in tables with some items in columns with headings and others individually labelled, in repeating paragraphs with many individual data items independently labelled, or in some other repeating structure with few (if any) labels describing the content, e.g., general search

results [Gregg and Walczak, 2007]. All of these variations in the structure of the web page and the definition of extraction targets can complicate the extraction process.

2.2 Extraction Technique

For a wrapper to extract data from a document, it needs to break it into tokens, apply extraction rules for each extraction target, and assemble the extracted values into individual records [Chang et al., 2006]. Generally, there are two classes of extraction rules that can be applied to web documents: extraction rules that rely on the position of the extraction target within the web page and extraction rules that rely on domain knowledge to locate and extract the target data.

2.3 Position based Extraction

Position based extraction relies on inherent structural features of HTML documents for accomplishing information extraction [see [Atzeni et al., 2002]. These systems use a parser to decompose an HTML document into a parse-tree that reflects its HTML tag hierarchy. Extraction rules are written to locate data based on this parse-tree hierarchy. The extraction rules can be either regular expression rules or Prolog like logic rules, which make an assignment between a variable name and a path expression. The primary limitation of position extraction is that when there are changes to the structure of the target page templates, the wrappers can fail to extract the desired information correctly. However, position wrappers do guarantee a high accuracy of information extraction, with both precision and recall being at least 98% [Chidlovskii, 2002]. In addition, it is possible to use wrapper induction to create position wrappers based on a sample of regularly formatted web pages, e.g., like those generated from a database using a web page template. Wrapper induction automatically builds a wrapper by learning from a set of sample pages [Crescenzi et al., 2001]. This can greatly speed the development and update of position based wrappers [Arasu and Garcia-Molina, 2003, Flesca et al., 2004, Kushmerick et al., 1997, Muslea and S. Minton, 1999].

2.4 Ontology based Extraction

An alternative to position extraction is to generate extraction rules based on knowledge about the reference domain [Embley et al., 1998, Seo et al., 2001]. Similar to position systems, ontology systems use wrappers which make an assignment between a variable name and specific domain key words used to label data within the document. They also use the lexical appearance of the data to help identify extraction targets. Since ontology extraction tools use domain knowledge to describe the data of interest, the wrappers generated using domain ontologies continue to work properly even if the formatting features of the source pages change, and will work for pages from many distinct sources

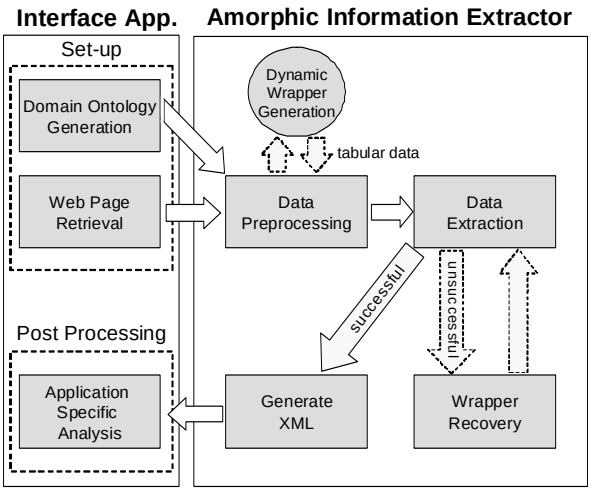

Figure 1: System architecture.

belonging to a same application domain [Embley et al., 1998]. One limitation of ontology extraction is that it requires the extractions targets to be fully described using page-independent features. This means all data to be extracted must either have unique characteristics or be labelled using context key words. Unfortunately, data of interest on the Web does not always meet these requirements.

3 The Amorphic System

The Amorphic system combines position extraction with ontology extraction to allow data to be extracted from a variety of web pages [Gregg and Walczak, 2007]. The system includes a wrapper recovery and repair module that allows it to recover from page structure/terminology changes that otherwise would cause the wrapper to fail. The current Amorphic system, shown in Figure 1, is designed to work in conjunction with a separate application interface that extracts information for applications with different extraction goals. On the front-end, the interface application provides tools to assist with the generation of domain specific ontologies and for the retrieval of web pages to be passed to the Amorphic information extraction system for processing. On the back-end, the interface application receives an XML file containing the structured tokens extracted from the web page and processes those tokens as needed for the current application.

Amorphic consists of several modules, namely: the data preprocessing module examines the page structure and determines how best to parse the site. It analyses the web page passed from the interface application, and uses the extraction rules to locate tokens of interest in the web page. The extraction rules supported by Amorphic include rules

```
<?xml version="1.0" ?>
- <DomainOntology>
  <Name>Item</Name>
- <ElementMap>
    <name>ItemID</name>
    <keyword>eBay item</keyword>
    <keyword>ID #:</keyword>
    <pattern>\d*\d</pattern>
    <type>String</type>
</ElementMap>
- <ElementMap>
    <name>Title</name>
    <keyword><title></keyword>
    <pattern>*\w*</pattern>
    <type>String</type>
</ElementMap>
  ...(omitted)
- <ElementMap>
    <name>NumBids</name>
    <keyword>History:</keyword>
    <keyword>Number of Bids:</keyword>
    <keyword># of Bids:</keyword>
    <pattern>\d*\d</pattern>
    <type>int</type>
</ElementMap>
  ...(omitted)
</DomainOntology>
```

Figure 2: Portion of the XML-based extraction rules for combined extraction.

that use regular expressions to specify the position of tokens within the web page as well as rules that use domain key words to locate data of interest within the web page (or a set of similar web pages from a variety of sites across a given domain). If the web page contains tabular data, a modified wrapper is generated that maps the table columns to tokens (domain specific key words) defined in the domain ontology. The data extraction module extracts the specific data from the web page. If the extraction completes successfully, it generates an XML file containing the extracted data and passes it back to the interface application; otherwise, the web page and domain ontology are passed to the wrapper recovery module, which attempts to locate the missing tokens and generate a revised domain ontology.

4 Extraction Resilience Experiment

The primary purpose of a resilient information extraction system is to adapt to changes in web sites over time without breaking, i.e., they are most useful in cases in which the web site (or domain) is subject to frequent structural changes that could cause position wrappers to fail. This section describes an experiment that compares the resilience of position extraction, ontology extraction, and the Amorphic combined approach.

4.1 Experiment Design

This study compares the data extracted from the eBay on-line auction site using the three extraction approaches. The eBay on-line auction site was selected for this experiment for several reasons. First, the eBay site has millions of individual web pages

Year	# Pages Parsed	% Data Items Retrieved (Combined)	% Data Items Retrieved (Position)	% Data Items Retrieved (Ontology)
2003	245	99.13%	98.58%	98.36%
2004	872	99.00%	75.92%	98.23%
2007	101	99.36%	14.28%	98.60%

Table 1: Information extracted from eBay over 4 years.

containing semi-structured data that can be extracted using either position or ontology extraction. Second, eBay changes the structure and content of the pages on their site frequently, requiring an extraction system that is resilient to these changes. During the period examined in the experiment eBay changed the layout of its web pages over 20 times, with 9 of the changes representing major changes that had a significant negative impact on position extraction rules. Finally, the eBay site contains data that is potentially interesting to application developers. Numerous researchers routinely use data from eBay as a part of their research, e.g., [Gregg and Scott, 2006, Gregg and Walczak, 2006b], and there are also specialised tools that rely on information extracted from this site to support on-line auction users, e.g., jBidWatcher.com).

Three wrappers were created to extract data from individual eBay listing pages. The first wrapper identified the content to be extracted based solely on its position within the web page. The second wrapper used domain specific key words and data patterns to locate the content to be extracted. The final wrapper used a combination of extraction rules. For example, the extraction rule for the auction title was specified using its position on the web page, since the auction title was unlabelled. The remaining data items (the Item ID, Closing Price, Staring Bid, Quantity, Number of Bids, Location, Start Time, Duration, End Time, Seller, Seller Rating, Buyer, Buyer Rating, Payment Information, Shipping Details, and Shipping Charge) were all extracted using ontology extraction rules since most were labelled and had a consistent format. A portion of the extraction rules used for the combined extraction is shown in Figure 2.

The experiment used the Amorphic information extraction tool to extract data from a sample of individual on-line auction pages on three separate occasions spanning a period of four years. The initial test in 2003 occurred immediately following the development of the three domain ontologies. The test in 2004 and the test in 2007 used the same wrappers as the initial test. As a part of each of these tests a sample of individual item pages were retrieved and the amount of information that was successfully extracted from the web pages was determined.

4.2 Results

The combined approach demonstrated superior performance to both the position wrapper and the ontology wrapper in all three tests, as shown in Table 1. In 2003, the three

information extraction wrappers were applied to 245 individual on-line auction listing pages. Since all three wrappers were developed immediately before the 2003 data collection, all three wrappers extracted data with an error rate of less than 2%. The position wrapper correctly extracted 98.58% of the data items of interest from the auction listing pages. The ontology wrapper correctly extracted 98.36% of the data of interest. The combined wrapper correctly extracted 99.13% of the data of interest. The combined wrapper extracted more data than either of the individual wrappers because the auction listings contain some data that position extraction rules could not locate (shipping charge data found in freeform text descriptions) and some data that ontology extraction rules had difficulty extracting (unlabelled auction titles). In July 2004 the resilience of the three information extraction approaches was evaluated by parsing an additional 872 auction listing pages. The combined wrapper extracted 99.0% of the data of interest and continued to demonstrate better performance than either the position wrapper (75.92%) or the ontology wrapper (98.23%). The resilience of the three information extraction approaches was evaluated for a third time by parsing 101 auction listing pages in August of 2007. The performance of the position wrapper degraded dramatically in the 2007 test with only 14.28% of the data of interest extracted correctly. This was due to major changes in the structure of the individual auction listing pages on eBay between 2003 (when the position wrapper was created) and 2007.

One of the surprising things about the experimental results is that the performance of both the ontology and the combined wrappers actually improved between 2004 and 2007: the precision of the ontology wrapper increased to 98.60%, and the precision of the combined wrapper increased to 99.36% in 2007; the reason is that eBay implemented a standardised shipping charge format between 2004 and 2007. The standardised shipping format used key words already found in the domain ontology, allowing both the combined and the ontology wrappers to more reliably locate and extract the shipping charge data.

5 Discussion and Conclusions

One of the biggest challenges when using web based information for organisational decision making is the unreliability of the Web as an information source. Web sites containing data of interest come in a wide array of formats, some include labels some do not, some are well structured and others are not. This presents a challenge for managing information extraction because frequently, a tool that can extract information from one site is not well suited to extracting data from another. A second challenge is that many commercial web sites frequently change the design of their sites requiring many wrappers written to work with the original site to fail or to become less effective. This presents a challenge to organisations interested in using web information for decision making because the effort required to maintain their information extraction wrappers often takes more time than the data is worth to the organisation.

This study demonstrates that ontology extraction is significantly more resilient than position extraction. Even after more than four years and many significant changes to the eBay web site, the wrappers using ontology extraction rules continued to have an error rate of less than 2%. However, the study also showed that there are some types of data that position extraction rules can extract more reliably than ontology extraction rules. The ability of the Amorphic information extraction system to locate data of interest based on domain-knowledge or page structure allows it to extract more data than the other two approaches. This combined system demonstrated good performance across the entire four year period evaluated in this article.

The experimental evaluation of the three information extraction approaches illustrates the need for information extraction systems capable of extracting information from semi-structured web documents. Contrary to the predictions of semantic web proponents, the majority of web sites do not seem to be moving towards labelling their content to facilitate processing by autonomous software agents. Instead, the past few years have led to an explosion of semi-structured human-readable data facilitated by Web 2.0 technologies like blogs, wikis and social network services. These sites can be potentially large information sources for information extraction applications. Additional studies of the different extraction approaches are needed to determine which types of extraction rules are best suited to these application domains. Studies could be undertaken at social network sites to examine comments to gain a more accurate estimate of the percent of young people at these sites that are approached by sexual predators or solicited to participate in some type of illegal activity. Other uses for data from these sites include applications that aggregate information on a wide variety of topics, e.g., FeedWiz [Schuff and Turetken, 2006].

There is also a need for additional research on information extraction systems, more specifically on improving the ability of systems to work with both position extraction and ontology extraction. For example, the ability to automatically create extraction rules for a combined position/ontology system using a set of sample pages needs to be improved. An extraction rule generation system needs to locate both potential key words and data of interest. The extraction rule generation system should also identify potential unlabelled extraction targets and add their position to the generated ontology rules.

References

[Arasu and Garcia-Molina, 2003] Arasu, A. and Garcia-Molina, H. (2003). Extracting structured data from web pages. *ACM SIGMOD Record*, 31(2):337–348.

[Atzeni et al., 2002] Atzeni, P., Mecca, G., and Merialdo, P. (2002). Managing web-based data: database models and transformations. *IEEE Internet Computing*, 6(4):33–37.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–43.

[Carroll and Klyne, 2004] Carroll, J. and Klyne, G. (2004). Resource Description Framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210.

[Chang et al., 2006] Chang, C., Kayed, M., Girgis, M., and Shaalan, K. (2006). Survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428.

[Chidlovskii, 2002] Chidlovskii, B. (2002). Automatic repairing of web wrappers by combining redundant views. In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, pages 399–406.

[Crescenzi et al., 2001] Crescenzi, V., Mecca, G., and Merialdo, P. (2001). RoadRunner: Towards automatic data extraction from large web sites. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 109–118.

[Embley, 2004] Embley, D. (2004). Toward semantic understanding: an approach based on information extraction ontologies. In *Proceedings of the 15th Australasian Database Conference*, pages 3–12.

[Embley et al., 1998] Embley, D., Campbell, D., Smith, R., and Liddle, S. (1998). Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 52–59.

[Embley et al., 1999] Embley, D., Jiang, Y., and Ng, Y. (1999). Record-boundary discovery in web documents. *ACM SIGMOD Record*, 28(2):467–478.

[Flesca et al., 2004] Flesca, S., Manco, G., Masciari, E., Rende, E., and Tagarelli, A. (2004). Web wrapper induction: a brief survey. *AI Communications*, 17(2):57–61.

[Gregg and Scott, 2006] Gregg, D. and Scott, J. (2006). The role of reputation systems in reducing on-line auction fraud. *International Journal of Electronic Commerce*, 10(3):97–122.

[Gregg and Walczak, 2006a] Gregg, D. and Walczak, S. (2006a). Adaptive web information extraction. *Communications of the ACM*, 45(5):78–84.

[Gregg and Walczak, 2006b] Gregg, D. and Walczak, S. (2006b). Auction advisor: on-line auction recommendation and bidding decision support system. *Decision Support Systems*, 41(2):449–471.

[Gregg and Walczak, 2007] Gregg, D. and Walczak, S. (2007). Exploiting the information Web. *IEEE Transactions on System, Man and Cybernetics Part C*, 37(1):109–125.

[Koivunen and Miller, 2002] Koivunen, M. and Miller, F. (2002). W3C Semantic Web activity. In *Proceedings of Seminar "Semantic Web Kick-Off in Finland"*, pages 27–44.

[Kushmerick et al., 1997] Kushmerick, N., Weld, D., and Doorenbos, R. (1997). Wrapper induction for information extraction. In *Proceedings of the of the International Joint Conference on Artificial Intelligence*, pages 729–735.

[Laender et al., 2002] Laender, H., Ribeiro-Neto, B., and A.S. da Silva, J. S. T. (2002). A brief survey of web data extraction tools. *ACM SIGMOD Record*, 31(2):84–93.

[Liu and Maes, 2005] Liu, H. and Maes, P. (2005). Interestmap: Harvesting social network profiles for recommendation. http://ambient.media.mit.edu/assets/_pubs/BP2005-hugo-interestmap.pdf.

[Muslea and S. Minton, 1999] Muslea, I. and S. Minton, K. (1999). A hierarchical approach to wrapper induction. In *Proceedings of the 3rd International Conference on Autonomous Agents*, pages 190–197.

[Sarawagi, 2002] Sarawagi, S. (2002). Automation in information extraction and integration. In *Proceedings of the 28th International Conference on Very Large Databases*. (Available from the authors).

[Schuff and Turetken, 2006] Schuff, D. and Turetken, O. (2006). FeedWiz: Using automated document clustering to map the blogosphere. In *Proceedings of the 4th Annual SIGDSS Pre-ICIS Workshop on Decision Support Systems*. (Available from the authors).

[Seo et al., 2001] Seo, K., Yang, J., and Choi, J. (2001). Building intelligent systems for mining information extraction rules from web pages by using domain knowledge. In *Proceedings of the IEEE International Symposium on Industrial Electronics*, pages 322–327.

[Turmo et al., 2006] Turmo, J., Ageno, A., and Català, N. (2006). Adaptive information extraction. *ACM Computing Surveys*, 38(2):1–47.