

Active Learning for Hierarchical Wrapper Induction

Ion Muslea, Steve Minton, and Craig Knoblock

Information Sciences Institute / University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90230, USA
{muslea, minton, knoblock}@isi.edu

Information mediators that allow users to integrate data from several Web sources rely on *wrappers* that extract the relevant data from the Web documents. Wrappers turn collections of Web pages into database-like tables by applying a set of extraction rules to each individual document. Even though the extraction rules can be written by humans, this is undesirable because the process is tedious, time consuming, and requires a high level of expertise.

As an alternative to manually writing extraction rules, we created STALKER (Muslea, Minton, & Knoblock 1999), which is a wrapper induction algorithm that learns high-accuracy extraction rules. The major novelty introduced by STALKER is the concept of *hierarchical* wrapper induction: the extraction of the relevant data is performed in a hierarchical manner based on the embedded catalog tree (ECT), which is a user-provided description of the information to be extracted. Consider the sample document

```
<html> Name: Joe's <p><br>Cuisine: American <p>  
Menu: Salad $2, Soup $1.5, Steak $4.25. </html>
```

It is easy to see that the document above has a hierarchical structure: at the top level, the whole page can be seen as a 3-tuple that contains the name, cuisine, and menu. The name and cuisine are atomic items (i.e., strings), while the menu is an *embedded list* of 2-tuples that contain the course name and the price. Consequently, the relevant data in the document can be seen as the leaves of a tree-like structure in which the root represents the whole page, and all internal nodes are embedded lists. STALKER generates one extraction rule for each node in the tree, together with an additional list iteration rule for each internal node.

Given the learned rules and the ECT of the documents, the extraction is performed in a hierarchical manner. A straightforward example would be to extract the restaurant name from the page above: we can use the rule `SkipTo(Name:)` to ignore everything until "Name:", which immediately precedes the restaurant name; then we can apply `SkipUntil(<p>)` to extract all characters until we find "<p>". In order to perform a more complicated task, say to extract the names of all the courses in the menu, STALKER first extracts the whole menu, and then it applies the corresponding list iteration rule to obtain the individual 2-tuples

that describe each course. Finally, the rule for the course name is applied to each 2-tuple obtained during the previous step.

Our approach has two main advantages. First, it can be applied to sources that contain arbitrarily many sibling and embedded lists. For instance, the pages might also include a list of accepted credit cards, and each 2-tuple in the menu might also include an embedded list of actual dishes (e.g., "Soup (bean, beef, chicken) \$1.5"). Second, as sibling nodes are extracted independently of each other, the learning process is not affected by the various orders in which the items may appear in the pages.

As labeling the training data is the major bottleneck in all inductive approaches to information extraction, researchers have tried to reduce the burden by using active learning (see (Califf 1998) and (Soderland 1999)). We created SGAL, which is a committee-based active learning algorithm that uses STALKER to generate 2-member committees of extraction rules. Our approach is similar to *active learning with committees* (Liere & Tadepalli 1997), except that our committee members are not chosen randomly: the two extraction rules in the committee belong to the most specific and most general borders of the version space (Mitchell 1977).

In this abstract we relate the idea of active learning with committees to version space, and we apply it to hierarchical wrapper induction. The initial results are promising: we compared SGAL and STALKER on 14 extraction tasks, and the former always does at least as well as the latter. More important, SGAL learns 100% accurate rules on four out of the five tasks on which STALKER fails to do so.

References

- Califf, M. 1998. Relational learning techniques for natural language information extraction. *PhD Thesis, U. Texas.*
- Liere, R., and Tadepalli, P. 1997. Active learning with committees for text categorization. *AAAI-97.*
- Mitchell, T. 1977. Version spaces: a candidate elimination approach to rule learning. *IJCAI-77.*
- Muslea, I.; Minton, S.; and Knoblock, C. 1999. A hierarchical approach to wrapper induction. *Auton. Agents-99.*
- Soderland, S. 1999. Learning information extraction rules for semi-structured and free text. *J. of Machine Learning.*