Gleaning answers from the web*

Nicholas Kushmerick
Computer Science Department, University College Dublin
nick@ucd.ie www.cs.ucd.ie/staff/nick

1 Introduction

This position paper summarizes my recent and ongoing research on Web information extraction and retrieval. I describe

- *wrapper induction and verification* techniques for extracting data from structured sources;
- *boosted wrapper induction*, an extension of these techniques to handle natural text;
- ELIXIR, our efficient and expressive language for *XML information retrieval*;
- techniques and applications for *text genre classification*; and
- stochastic models for *XML schema alignment.*

The unifying theme of these various research projects is to develop enabling technologies that facilitate the rapid development of large Web services for data access and integration.

2 Wrapper induction and verification

A wide variety of valuable textual information resides on the Web, but very little is in a machine-understandable form such as XML. Instead, the content is usually embedded in HTML markup or other encodings designed for human consumption. The information extraction task is to automatically populate a database with content gleaned from information sources such as Web pages.

Wrappers are an important special case of the general information extraction task. A wrapper is a specialized information extraction module tailored for a particular source. For example, a meta-search engine needs a distinct wrapper for each of its underlying search engines.

Wrappers are usually fairly simple pattern-matching programs, because in many applications the documents being processed are highly regular, such as machine-generated HTML text emitted from CGI scripts. Nevertheless, automated approaches to wrappers construction are essential if we want to scale up our applications to integrate data from dozens or thousands of sources.

Wrapper induction [8, 12, 10] involves using machine learning techniques to automatically generate wrappers. The input to a wrapper induction algorithm is a set of training examples (sample documents annotated with the information that should be extracted from each), and the output is a wrapper. My contributions include the identification of several classes of wrappers that are both:

- sufficiently expressive to wrap numerous real Web sources (70% of the data-intensive sites could be handled by one of the wrapper classes, according to a survey); and
- efficiently learnable (accurate wrappers can be learned from a handful of training examples, and the learning algorithm consumes only a fraction of a CPU second per example).

While wrapper induction is an important first step, most research ignores an important complication. Since applications that consume extracted data rarely have control over the remote sources, a deployed wrapper may suddenly fail if the formatting regularities on which it relies are violated.

While automatically repairing a wrapper is the ultimate goal, it is challenging simply to determine whether a wrapper is operating correctly. The problem is there are two moving targets: the formatting, or the content itself, or both, may change, and a wrapper verifier must distinguish between these two types of change. I have developed a wrapper verification algorithm [9, 11] that uses training data to learn a probabilistic model of the data correctly extracted by a wrapper. To verify a wrapper, its output is evaluated against the model to estimate the probability

*Position paper, AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases.

that the wrapper is operating correctly.

3 Boosted wrapper induction

While the wrapper induction algorithms were designed for regularly formatted documents such as machine-generated HTML, we wondered whether the approach could be extended to natural text such as newspaper articles or email messages. The extraction rules learned by wrapper induction are simple contextual patterns such as "extract '`<b>`*anything*`</b>`' as a telephone number". Can information be successfully extracted from natural text by composing many such simple rules (eg, "extract '`call me at` *anything* `before`' or '`phone:` *anything* `(home)`' as a telephone number")?

To explore this possibility, we used boosting, a machine learning technique for improving the accuracy of learning algorithms with high accuracy but low coverage, such as the simple wrapper induction algorithms. The result is the boosted wrapper induction algorithm [7], which is competitive at numerous real-world extraction tasks with state-of-the-art algorithms (including hidden Markov models and several rule-learning algorithms), and superior in many.

A demonstration of boosted wrapper induction is available at www.smi.ucd.ie/bwi.

4 XML information retrieval

XML has emerged as a powerful standard for the interchange of structured documents. Given its structured nature, XML has given rise to new languages for querying, traversing, filtering and transforming XML content.

These languages have been developed by database rather than information retrieval researchers, so the languages do not support ranked or graded queries in the spirit of traditional information retrieval. For example, one can use existing query languages to retrieve sections of an XML document containing exactly the phrase "engineering reports", but it is impossible to rank these results by the degree to which they satisfy the query (e.g., sections that contain just one of the terms, or variants such as "engineer" or "reporting" should be retrieved at lower rank).

We fill this gap with ELIXIR [3, 1, 2], an efficient and expressive language for XML information retrieval. ELIXIR extends existing XML query technology with a graded textual similarity operator. Unlike related efforts, our language efficiently supports similarity joins, an essential capability for data integration applications with noisy textual identifiers. For example, to identify possible duplicates in a list of book titles, one could use an ELIXIR similarity join to identify pairs of titles that are mutually similar.

A demonstration of ELIXIR is available at www.smi.ucd.ie/elixir.

5 Text genre classification

Most text classification research has focused on *objective* classes reflecting document topics or concepts, such as classifying newspaper articles into 'financial', 'sports', etc. In contrast, we are examining subjective *genre* classification tasks such as whether a newspaper article expresses 'opinions' as opposed to 'facts', whether a product review is 'positive' or 'negative', whether an article is 'detailed' or 'superficial', the amount of technical knowledge assumed by a document's author, etc.

Existing text classification algorithms exploit words that reliably indicate the correct class. Our investigation [5, 6, 4] suggests that such term-based techniques are inaccurate for our subjective tasks. Instead, we use shallow natural language processing techniques such as part-of-speech tagging and estimates of term topicality to derive document features that yield accurate classification. Our experiments in the newspaper domain confirm that part-of-speech tags yield significantly better performance than term features.

We our employing our techniques in two applications: information filtering in heterogeneous digital libraries, and the visualization of information retrieval search results.

A demonstration of our text classification techniques is available at www.smi.ucd.ie/hyppia.

6 Stochastic models for XML schema integration

The well-known database schema integration challenge is that data retrieved from heterogeneous sources can not be integrated without a mapping from the remote schemas to some global or mediated schema.

We focus on one particular task: automatically constructing mappings between heterogeneous sources of semi-structured XML data. For example, if you export your Rolodex data with `<name>` and `<address>` tags, but I use `<jmeno>` and `<ulice>`, our data can't be integrated without knowing that `<name>`=`<jmeno>` and `<address>`=`<ulice>`. This problem has received some attention recently, but these efforts largely ignores the constraints imposed by XML's nested structure.

To address this problem, we are developing a novel stochastic finite-state model called the Stochastic Document Markup Model. SDMMs are similar to hidden Markov models: they contain states linked by stochastic transitions, and states emit tokens according to a given distribution. Some states (the "tag" states) emit XML tags such as <name> or <jmeno>, while others emit the actual element contents (eg, `12 Main St`).

An essential difference between SDMMs and HMMs is that at decoding time we do not know the tag states' emission distributions. For example, we know that a given state emits an XML tag for "name"—but we don't know what that tag is! Furthermore, in decoding, we demand a path that is both cyclic (ensuring only well-formed XML is generated) and consistent (eg, if the "name" state emits <jmeno>, then no other tag can be emitted subsequently). Since SDMM decoding appears harder than for HMMs, we are examining heuristic approximations, such as finding a path with probability within a constant factor of optimal, or paths that are "almost" consistent or properly nested.

Acknowledgments. This research was funded by grants from Enterprise Ireland, the US Office of Naval Research, the US National Science Foundation, and a University College Dublin President's Research Award. As listed in the bibliography, I am grateful to my many colleagues at the University of Washington and University College Dublin.

References

[1] T. Chinenyanga and N. Kushmerick. An efficient and expressive language for XML information retrieval. *Journal of the American Society for Information Science and Technology*, 2001. In press.

[2] T. Chinenyanga and N. Kushmerick. Expressive and efficient ranked querying of XML data. In *Proc. ACM SIGMOD Workshop on the Web and Databases*, 2001.

[3] T. Chinenyanga and N. Kushmerick. Expressive retrieval from XML documents. In *Proc. ACM Int. Conf. Research and Development in Information Retrieval*, pages 163–171, 2001.

[4] M. Dimitrova, A. Finn, N. Kushmerick, and B. Smyth. Web genre visualization. 2002. Submitted to *Conference on Human Factors in Computing Systems*.

[5] A. Finn, N. Kushmerick, and B. Smyth. Fact or fiction: Ccontent classification for digital libraries. In *Proc. NSF/DELOS Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.

[6] A. Finn, N. Kushmerick, and B. Smyth. Genre classification and domain transfer for information filtering. In *Proc. European Colloqium on Information Retrieval Research*, 2002.

[7] D. Freitag and N. Kushmerick. Boosted wrapper induction. In *Proc. 17th Nat. Conf. AI*, pages 577–583, 2000.

[8] N. Kushmerick. *Wrapper Induction for Information Extraction*. PhD thesis, Univ. of Washington, 1997.

[9] N. Kushmerick. Regression testing for wrapper maintenance. In *Proc. 16th Nat. Conf. AI*, pages 74–79, 1999.

[10] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *J. Artificial Intelligence*, 118(1–2):15–68, 2000.

[11] N. Kushmerick. Wrapper verification. *World Wide Web Journal*, 3(2):79–94, 2000.

[12] N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Proc. 15th Int. Joint Conf. AI*, pages 729–735, 1997.