

Evaluating the Role of Context in Syntax Directed Compression of XML Documents

S. Hariharan

Priti Shankar¹

Department of Computer Science and Automation, Indian Institute of Science,
Bangalore 560012, India.

We propose a new technique based on recursive finite state machines, for tracking context to be used in a statistical code compression scheme for XML documents. An arithmetic coding scheme [1] is used for this purpose. We also study the tradeoffs between space and compression ratio, by observing the effects of either using or ignoring root to leaf contexts for textual content in the associated tree structures. The advantage of our scheme is that it is syntax aware, and the compressor and decompressor can be generated automatically from the Document Type Definition(DTD) without interactive inputs from the user. The automaton that is constructed mirrors the DTD, in that it tracks the structure of the document, and is able to make accurate predictions of expected elements. Therefore whenever the predicted element is unique, there is no need to encode it at all, as the decoder generates the same automaton from the DTD and is thus able to generate the unique expected symbol. Most markup symbols fall into this category of symbols. Character data associated with a single element can either be automatically directed to the same model for arithmetic compression irrespective of the *instance* of the element in the DTD, in which case the model is said to be *path agnostic*, or one may choose to have a separate model for each root to leaf path in the underlying tree for the document, in which case the scheme is *path sensitive*. We evaluate both schemes in this paper. Since elements may be nested recursively, the device used in general, is a set of *mutually recursive* automata. A stack is used to store root to leaf context in the underlying structure tree, and operations on the stack are governed by syntax. An instantaneous description of the device completely defines the appropriate model to be used. We have measured the effects of allocating a fixed size block of runtime memory for the compressor, as well as varying strategies for flushing out context tables. We have also compared the path sensitive and path agnostic schemes for storing context for PCDATA. Our experiments indicate that path sensitive schemes are less effective in the fixed memory model. Our experiments are run on some massive databases and we compare the performance of our tool with that of the XML conscious tool XMLPPM [2].

References

- [1] Ian H. Witten, Radford M. Neal, John G. Cleary.: Arithmetic Coding for Data Compression. *Communications of the ACM*, 30(6): 520-540, June 1987.
- [2] James Cheney.: Compressing XML with Multiplexed Hierarchical Models. *Proceedings of the 2001 IEEE Data Compression Conference*, pp. 163-172.

¹Contact Author: priti@csa.iisc.ernet.in