# XML Compression using Hierarchical Multiplexed PPM Models

James Cheney

Cornell University

Data Compression Conference

Snowbird, Utah

March 27, 2001

# What is XML?

XML is...

- a recent (1998) standard for Web data
- a tree structured data language: nested elements, attributes, text
- a bandwagon: Microsoft, Sun, IBM on board

# XML Markup is Verbose

Terseness was *not* a design goal

```
<ATOM>
  <NAME>Actinium</NAME>
  <ATOMIC_WT>227</ATOMIC_WT>
  <ATOMIC_NO>89</ATOMIC_NO>
  <BOIL_PT UNITS="Kelvin">3470</BOIL_PT>
  <SYMBOL>Ac</SYMBOL>
  ...
</ATOM>
```

“XML bloat”: redundant tags, syntax

Concern that this will limit practical use of XML

# XML compression = text compression?

Writing compressors is hard work

XML is basically text, so use existing compressors

Transform the document reversibly, then compress

XMILL [Liefke, Suciu 2000]

XMLZIP [XML Solutions]

This seems easier, but...

Can fully XML-conscious compressors do better?

# In this talk

- Baseline: BZIP2 on XML text
- New transformation technique: Encoded SAX [ESAX]

  Encodes sequence of XML parsing events
- New approach: Multiplexed Hierarchical Modeling [MHM]

  Uses tree-structure context in multiplexed PPM models

Result: 5–20% better XML compression than baseline

# Baseline

Compression experiments

Built an XML compression corpus

Compressed using GZIP, BZIP2, PPM, PPM*

Experimented with XMILL, XMLZIP

Bottom line

BZIP2 on XML text compresses best

# Encoded SAX

Idea:

Encode element and attribute names using symbol tables

Then compress using BZIP2

Results:

ESAX+BZIP2: 2–3% better than BZIP2 alone

Can we do better?

# Idea 1: Multiplexing

Idea:

Split data into 4 interleaved streams:

Elements, Attributes, Text, Symbols

Use different PPM model for each stream

Why it helps:

Streams share symbols but behave differently

Multiplexing avoids stream averaging

# Idea 2: Hierarchical context

Idea:

Leak "ghost" element context symbols into streams

Ghost symbols are scanned but not encoded or decoded

Why it helps:

Element context often relevant: numbers vs. text

Ghost symbols cut down on "context switch" penalty

# Implementation

`xmlppm`: an implementation of MHM

Used PPMD+, PPM* implementations [Teahan,Moffat]

Implementation online at `http://xmlppm.sourceforge.net`

# Results

Using PPM models:

compresses documents about 5% better

about as fast as PPM

Using PPM* models:

compresses about 8% better overall

compresses about 20% better on highly structured data

about as slow as PPM*

# Conclusions

- ESAX+BZIP2

  compresses fast, 2–3% better than baseline

- MHM using PPM

  slower, but about 5% better than baseline overall

- MHM using PPM*

  very slow, but about 20% better on structured XML

MHM compresses XML documents substantially better than existing solutions

# Supervised vs. unsupervised

Unsupervised: compressor on its own

GZIP, XMLZIP, default XMILL

Supervised: compressor has human help

XMILL path expression language

This talk/paper: *unsupervised* only