Improving HTML compression

Przemysław Skibiński
University of Wrocław, Institute of Computer Science,
Joliot-Curie 15, 50-383 Wrocław, Poland, e-mail: inikep@ii.uni.wroc.pl

Nowadays the Hyper Text Markup Language (HTML) is a standard for Internet web pages. HTML has many advantages, but its main disadvantage is verbosity, which can be coped with by applying data compression. HTTP protocol already supports Deflate (gzip) compression of HTML data, but Deflate is a general-purpose compression algorithm and much better results can be achieved with a compression algorithm specialized for dealing with HTML documents.

The primary objective of our research was to design an efficient way of compressing HTML documents, which will reduce Internet's traffic or will reduce storage requirements of HTML data. In our work we present the Lossless HTML Transform (LHT) aiming to improve lossless HTML compression in combination with existing general purpose compressors. The main components of our algorithm are: a static dictionary or a semi-static dictionary of frequent alphanumerical phrases, and binary encoding of popular patterns, like numbers, dates or IP addresses. Alphanumerical phrases are not limited to "words" in a conventional sense as they can be XML tags, XML entities, URL addresses, e-mails, and runs of spaces.

We have developed two versions of LHT: static and semi-static. Both algorithms have some disadvantages. Static LHT uses a fixed English dictionary required for compression and decompression. Semi-static LHT does not support streams as input (offline compression) as it requires two passes over an input file. Semi-static LHT creates a dictionary in a first pass and stores it within the compressed file.

LHT could be combined with any general-purpose compression algorithm, but we chose two algorithms of this kind: Deflate (well known from zip, gzip, and plenty of other applications) and PPMVC (which gives very good compression effectiveness in mediocre compression time and memory requirements), employing the same algorithms as the final stage of LHT to demonstrate the improvement from applying the HTML transform. We have also tried to use SCMPPM [1], a compressor for semi-structured documents, but it doesn't work with almost all HTML files that we used for experiments. On the remaining files it has compression effectiveness similar to PPMVC without LHT.

For experiments we have used HTML files (without images, etc.) downloaded from Internet, mostly from the following web pages: http://groups.google.com/, http://www.cs.fit.edu/~mmahoney/compression/, http://www.arturocampos.com/. The size of files span from 5 up to 170 kB.

Compared to the general-purpose compression algorithms, the static LHT improves HTML compression on average by over 17% in case of Deflate and almost 8% in case of PPMVC. The semi-static LHT improves HTML compression on average by almost 5% in case of Deflate and there is only a slightly improvement in case of PPMVC.

LHT has many nice practical properties. The transform is completely reversible (white characters are also preserved), the decoded document is an accurate copy of the input document. Moreover, LHT is implemented as a stand-alone program, requiring no external compression utility, HTML parser, thus avoiding any compatibility issues.

REFERENCES

[1] Joaquín Adiego, Gonzalo Navarro, and Pablo de la Fuente. Using Structural Contexts to Compress Semistructured Text Collections. *Information Processing and Management (IPM)* 43:769-790, 2007.

1068-0314/08 $25.00 © 2008 IEEE
DOI 10.1109/DCC.2008.74