ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CNTT & TRUYỀN THÔNG

HÀ DIỆU THÚY

NÉN DỮ LIỆU TIẾNG VIỆT SỬ DỤNG PHƯƠNG PHÁP MÃ HÓA SỐ HỌC

Chuyên ngành: Khoa học máy tính

Mã số: 60 48 01

TÓM TẮT LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Thái Nguyên – 2013

Công trình được hoàn thành tại
TRƯỜNG ĐẠI HỌC CNTT & TRUYỀN THÔNG

Người hướng dẫn khoa học: **PGS.TS Nguyễn Hữu Điển**

Phản biện 1: TS. Lê Quang Minh

Phản biện 2: TS. Trần Đức Sự

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn họp tại:
Trường Đại học Công nghệ thông tin & Truyền thông

Vào hồi 11 giờ 00 ngày 09 tháng 11 năm 2013

Có thể tìm hiểu luận văn tại:

- Trung tâm học liệu Đại học Thái Nguyên
- Thư viện trường Đại học CNTT & Truyền thông Thái Nguyên

MỞ ĐẦU

1. Đặt vấn đề

Nén dữ liệu là một kỹ thuật quan trọng trong rất nhiều lĩnh vực khác nhau. Chính nhờ có kỹ thuật nén dữ liệu mà ngày nay chúng ta có những phương tiện truyền thông hiện đại phục vụ cho cuộc sống như truyền hình cáp, truyền hình số, điện thoại, internet, các hệ thống lưu trữ, văn bản ... và rất nhiều khía cạnh khác. Do đó kỹ thuật nén dữ liệu ngày càng được quan tâm và phát triển nhiều hơn.

Tiếng Việt là một ngôn ngữ thuộc hệ thống chữ cái Latinh, sử dụng nhiều dấu đi kèm với nguyên âm, ngoài bảng chữ cái của tiếng Anh, tiếng Việt còn có thêm các ký tự:

Sáu nguyên âm a, e, i, o, u, y với 5 dấu thanh (sắc, huyền, hỏi, ngã, nặng) tổ hợp thành 30 ký tự.

Sáu nguyên âm ă, â, ê, ô, ơ, ư với sáu dấu thanh (sắc, huyền, hỏi, ngã, nặng, không dấu) tổ hợp thành 36 ký tự.

Một phụ âm đặc biệt đ.

Vậy cần thêm (30 + 36 +1) x 2 = 134 ký tự cho tiếng Việt.

Với bảng mã ASCII 8 bit sử dụng phổ biến trên máy tính, chúng ta có thể mã hóa 256 ký tự. Tuy nhiên, các ký tự có mã từ 0 đến 127 đã được chuẩn hóa và thuộc diện “cấm vi phạm” vì vậy chỉ còn 128 chỗ (mã từ 128 đến 255) là được “tự do”. Vậy nếu xây dựng mỗi chứ ứng với một mã thì sử dụng hết vùng tự do mà vẫn thiếu 134 – 128 = 6 chỗ.

Hiện nay chúng ta đang sử dụng chuẩn Unicode để lưu trữ các ký tự tiếng Việt. Như chúng ta biết chuẩn Unicode là chuẩn 2byte, do vậy khi lưu trữ các văn bản tiếng Việt trên các hệ thống lưu trữ sẽ xẩy ra tình trạng dư thừa dữ liệu. Điều này dẫn đến việc lưu trữ và xử lý sẽ lãng tài nguyên hệ

thống, khi truyền tải trên các đường truyền mạng sẽ chiếm băng thông nhiều hơn. Từ các yêu cầu thực tế đó đòi hỏi chúng ta phải loại bỏ sự dư thừa dữ liệu đó trước khi lưu trữ và xử lý. Chính vì thế em chọn đề tài "***Nén dữ liệu tiếng Việt sử dụng thuật toán mã hóa số học"***

2. Đối tượng và phạm vi nghiên cứu

- Các chuẩn lưu trữ tiếng Việt (Unicode, TCVN3, VNI-Vindows…)

- Các phương pháp và kỹ thuật nén dữ liệu

- Các phần mềm nén dữ liệu hiện nay

3. Hướng nghiên cứu đề tài

- Nghiên cứu các phương pháp nén dữ liệu như nén bảo toàn dữ liệu (lossless data compression) và nén mất mát dữ liệu (lossy data compression).

- Nghiên cứu các kỹ thuật nén dữ liệu như: kỹ thuật xử lý sự lặp lại của xâu (RLE), mã hóa Huffman, kỹ thuật nén LZW (Lempel - Zip và Welch)…

- Nghiên cứu về kỹ thuật nén bảo toàn dữ liệu Arithmetic Coding (Phương pháp mã hóa số học).

- Cài đặt thực nghiệm việc nén dữ liệu bằng Arithmetic Coding.

- Phân tích, so sánh và đánh giá kết quả thực nghiệm với các kỹ thuật nén dữ liệu (văn bản tiếng Việt) khác.

4. Phương pháp nghiên cứu

- Nghiên cứu các tài liệu về các kỹ thuật mã hóa và nén dữ liệu.

- Tìm hiểu các chuẩn tiếng Việt hiện nay ở Việt Nam.

- Khảo sát thực tế các phần mềm nén dữ liệu hiện nay đối với việc nén các văn bản tiếng Việt.

- Phân tích, đánh giá các kỹ thuật (thuật toán) nén dữ liệu.

- Cài đặt kỹ thuật nén Arithmetic Coding

- Triển khai thử nghiệm trên các loại dữ liệu văn bản tiếng Việt.

5. Ý nghĩa khoa học và ý nghĩa thực tiễn của đề tài

- Nghiên cứu hoàn thiện các kỹ thuật nén bảo toàn dữ liệu cho các văn bản tiếng Việt.

- Xây dựng ứng dụng nén dữ liệu cho các văn bản tiếng Việt.

Chương 1: TỔNG QUAN VỀ NÉN DỮ LIỆU

1.1. Tổng quan về nén dữ liệu

1.1.1. Sơ lược về nén dữ liệu

1.1.1.1 Khái niệm nén dữ liệu

Nén dữ liệu là quá trình làm giảm lượng thông tin "dư thừa" trong dữ liệu gốc và do vậy, lượng thông tin thu được sau nén thường nhỏ hơn so với dữ liệu gốc rất nhiều.

1.1.1.2 Nguyên tắc nén dữ liệu

Thông thường, hầu hết các tập tin trong máy tính có rất nhiều thông tin dư thừa, việc thực hiện nén tập tin thực chất là mã hoá lại các tập tin để loại bỏ các thông tin dư thừa.

Nhìn chung không thể có phương pháp nén tổng quát nào cho kết quả tốt đối với tất cả các loại tập tin vì nếu không ta sẽ áp dụng n lần phương pháp nén này để đạt được một tập tin nhỏ tuỳ ý. Kỹ thuật nén tập tin thường được áp dụng cho các tập tin văn bản (Trong đó có một số kí tự nào đó có xác suất xuất hiện nhiều hơn các kí tự khác), các tập tin ảnh bitmap (Mà có thể có những mảng lớn đồng nhất), các tập tin dùng để biểu diễn âm thanh dưới dạng số hoá và các tín hiệu tương tự (analog signal) khác (Các tín hiệu này có thể có các mẫu được lặp lại nhiều lần). Đối với các tập tin nhị phân như tập tin chương trình thì sau khi nén cũng không tiết kiệm được nhiều.

Ngoài ra, trong một số trường hợp để nâng cao hệ số nén người ta có thể bỏ bớt một số thông tin của tập tin.

1.1.2. Các phương pháp nén dữ liệu

1.1.2.1 Nén bảo toàn dữ liệu

Đó là mô hình nén dữ liệu mà nó cho phép người sử dụng bảo toàn thông tin trong suốt quá trình nén. Điều này được giải thích như sau:

Giả sử ta có dữ liệu nguồn là A và dữ liệu nén là A'. Sau khi ta giải nén A' thì được tập A" mà tập A" hoàn toàn giống với tập A ban đầu khi được giải nén. Thông thường, kỹ thuật này được áp dụng với các loại dữ liệu như văn bản vì độ chính xác của văn bản.

1.1.2.2 Nén hao hụt dữ liệu

Trong kỹ thuật nén, bên cạnh nén bảo toàn thì người ta còn đưa ra khái niệm nén không bảo toàn (hay còn gọi là nén hao hụt dữ liệu). Nén không bảo toàn là mô hình nén dữ liệu mà tính bảo toàn của dữ liệu không được coi trọng. Nó có nghĩa là nếu ta có tập dữ liệu A, tập nén A' thì sau khi giải nén ta thu được tập A" khác tập A ban đầu. Kỹ thuật này thường áp dụng cho việc nén dữ liệu là các loại tệp ảnh vì nói chung nó cũng không ảnh hưởng gì nhiều đến hình dạng ảnh.

1.2 Các kỹ thuật nén dữ liệu văn bản

1.2.1 Xử lý lặp lại của xâu ký tự (Run – Length Encoding)

Mục đích của thuật toán là tìm ra được ký tự lặp lại nhiều lần và số lần lặp lại của ký tự đó, thay thế cụm lặp lại bằng một biểu diễn nhỏ gọn hơn.

Có thể biểu diễn rút gọn dưới dạng cặp 3 ký hiệu (r,s,l) với s là ký hiệu của một dãy các ký tự nằm trong bảng chữ cái, r và l sẽ là các ký hiệu không được xuất hiện trong bảng chữ cái, tùy từng trường hợp mà r và l lại có ý nghĩa khác nhau.

Ví dụ: Cho chuỗi đầu vào là MMMMMMM (chữ M được lặp lại 7 lần), chuỗi này có thể thay thế bằng (r,7,M) hay viết tắt là r7M. r với ý nghĩa ký hiệu cho việc xuất hiện sự lặp lại (repeating) đòi hỏi chữ r không được xuất hiện trong bảng chữ cái của đầu vào.

Cho chuỗi đầu vào là ABCDEFG (không xuất hiện sự lặp lại) chuỗi này có thể thay thế bằng (n,7,ABCDEFGH), hay viết tắt là n7ABCDEFG. n

với ý nghĩa là ký hiệu cho việc không xuất hiện sự lặp lại (non-repeating), chữ n không được xuất hiện trong bảng chữ cái đầu vào.

Thuật toán này hiệu quả nếu như dữ liệu đầu vào gồm nhiều ký tự bị lặp lại liên tiếp. Ký tự đầu vào có thể ở dạng chữ trong bảng chữ cái, có thể là các bit 0, 1 nhị phân; các thông số về màu của các điểm ảnh, cũng có thể là các khối hợp thành của dữ liệu kiểu âm thanh. Trên thực tế, thuật toán này vẫn còn được áp dụng cho tới ngày nay: thuật toán HDC (hardware data compression), được sử dụng trong các ổ băng kết nối với hệ thống máy tính IBM, và cả thuật toán tương tự được dùng trong chuẩn SNA (System network architecture) của IBM.

1.2.2 Mã hóa Huffman

1.2.2.1. Mã Huffman tĩnh

* Nguyên lý:

Nguyên lý của phương pháp Huffman là mã hoá các bytes trong tệp dữ liệu nguồn bằng biến nhị phân. Nó tạo mã độ dài biến thiên là một tập hợp các bits. Đây cũng là một phương pháp nén kiểu thống kê, những ký tự xuất hiện nhiều hơn sẽ có mã ngắn hơn.

Mã Huffman có một tính chất quan trọng: mã của một ký hiệu này không thể là phần đầu của mã một ký hiệu khác.

Nếu như một ký hiệu được mã hoá bằng tổ hợp nhị phân 101 thì tổ hợp 10110 không thể là mã của một ký hiệu khác trong tệp nguồn. Do đó khi giải mã cần phải đọc lần lượt các bit cho đến khi gặp mã của ký hiệu nào đó.

* Thuật toán:

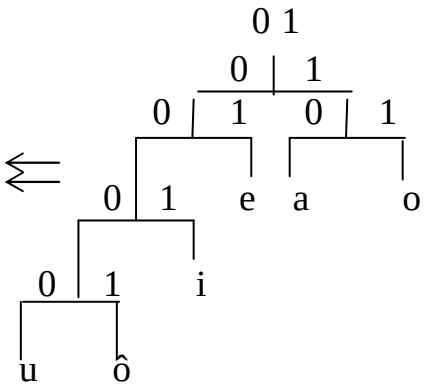
Việc xây dựng cây mã hoá Huffman được tiến hành bởi một thuật toán khác với thuật toán Fano - Shannon. Nếu như cây Fano - Shannon được xây dựng từ trên xuống dưới bằng cách chia đôi và gán cho mỗi

phần 1 bít, công việc kết thúc khi không thể tiến hành phân chia tiếp thì cây Huffman lại được thiết kế từ dưới lên, bắt đầu từ các lá của cây và công việc kết thúc tại điểm gốc.

Ví dụ :

Cho mô hình nguồn có các trạng thái và tần suất tương ứng như sau: (A, 0.2); (E, 0.3); (I, 0.1), (0. 0,2); (U, 0.1); (Ô, 0.1) ta có:

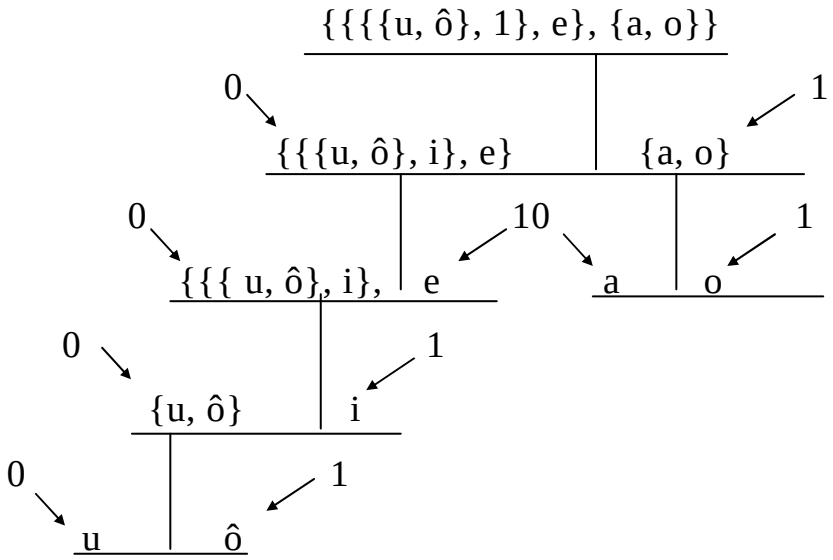
Ký tự	Tần xuất	Mã
A	0.2	10
E	0.3	01
I	0.1	001
O	0.2	11
U	0.1	0000
Ô	0.1	0001



Bước1: Nhóm 2 chữ cái có tần suất nhỏ nhất tạo ra chữ cái kép. Sau mỗi lần nhóm số chữ cái ít đi 1.

c -> 0.3	e -> 0.3	e -> 0.3	{a, 0} -> 0.4{{{u, ô},:}
a -> 0.2	a -> 0.2	{{u,ô},i} -> 0.3	e -> 0.3 {a, o{ ->
o -> 0.2	o -> 0.2	a -> 0.2	{{u, ô}, i} -> 0.3
i -> 0.1	{u,ô} -> 0.2	o -> 0.2	
u -> 0.1	i -> 0.1		
ô -> 0.1			

Bước 2: Tạo cây phân nhánh ngược với quá trình nhóm từ nhánh trái có mã 0, nhánh phải mã 1.



Vậy mã của ký tự là: u -> 0000; e -> 01

ô -> 0001; a -> 10

i -> 001; o -> 11

Thuật toán nén:

Bước 1: Tìm hai ký tự có trọng số nhỏ nhất ghép lại làm một, trọng số của ký tự mới bằng tổng trọng số của hai ký tự đem ghép.

Bước 2: Trong khi số lượng ký tự trong danh sách còn lớn hơn một thì thực hiện bước một, nếu không thì thực hiện bước ba.

Bước ba: Tách ký tự cuối cùng và tạo cây nhị phân với qui ước bên trái mã 0, bên phải mã 1.

Thuật toán giải nén:

Bước 1: Đọc lần lượt từng bit trong tập tin nén và duyệt cây nhị phân đã được xác định cho đến khi hết một lá. Lấy ký tự ở lá đó ghi ra tệp giải nén.

Bước 2: Trong khi chưa hết tập tin nén thì thực hiện bước một, ngược lại thực hiện bước ba.

Bước 3: Kết thúc thuật toán.