COMP9319 Web Data Compression and Search

Course Overview,
Information Representation &
Background

1

1

What is COMP9319 ?

- **how** different compression tools work.
- **how** to manage a large amount of text data (on small devices or servers).
- **how** to search gigabytes, terabytes or petabytes of data.
- **how** to perform full text search efficiently with heavy indexing, light indexing / no indexing.
- optional advanced topics (if time allows): distributed repositories, cloud etc.

2

2

Course Aims

As the amount of Web data increases, it is becoming vital to not only be able to search and retrieve this information quickly, but also to store it in a compact manner. This is especially important for mobile devices which are becoming increasingly popular. Without loss of generality, within this course, we assume Web data (excluding media content) will be in XML and its like (e.g., HTML, JSON).

This course aims to introduce the concepts, theories, and algorithmic issues important to Web data compression and search. The course will also introduce the most recent development in various areas of Web data optimization topics, common practice, and its applications.

3

3

Course info

- Homepage: www.cse.unsw.edu.au/~cs9319
- Lectures:
  - Wed 12:00-14:00 (live online lecture)
  - Fri 15:00-17:00 (pre-recorded)
  - Weeks 1-5, 7-10 (flexi week 6: no classes)

4

4

Lecturer in charge

Raymond Wong
Office: K17 Level 2 (Room 213)
Email: wong@cse.unsw.edu.au
Ph: 90659925
www.cse.unsw.edu.au/~wong

5

5

Lectures

For 2022T2:

- Live lecture (via Collaborate) on Wed: 1-2hrs
- Pre-recorded topic-based lectures on Fri: 2-3hrs

- *Live lectures will be recorded as well (in Moodle's Echo360)*

6

6

Recorded Topic-based Lectures

- Go through the scheduled topics in details
- Less problematic due to bad connections (from your side or my side)
- Less interruption due to Q&A
- **Note: we assume that you will watch the recordings every week; and attend & ask any questions at the live lectures / "consultations"**

7

7

Live lectures

- Topic-based recordings are good but no interactions or Q&A
- Hence there is a 1-hour live lecture (Wed 12pm, may go slightly overtime to 1-2hr if needed) to go through more examples, and/or answer any Q&A for the topics covered from the prev. wk.

8

8

Exercises

Exercises will be provided on WebCMS regularly.

- Brief solutions will be released one week later
- If you're stuck, please talk to us in the consultations.

9

9

No tutorials

- Have consultation slots instead
- Lecture Q&A will take place during the Collaborate live lectures
- Specific lecture / exercise / assignment questions can be addressed better in consultations
- Don't leave your questions till very late, we won't be able to address questions that stacked for many weeks

10

10

Consultations

- Week 2 - **Week 11** (including wk 6)
- 3 days a week
  - ➔ So please utilize them.
- Run in a hybrid consultation/tut style
- Discuss any questions on exercises & assignments
- Provide assistance on lecture materials

11

11

Consultations

Consultations

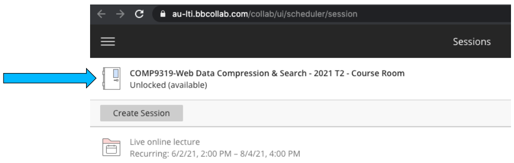
Day	Start Time	End Time	Room	Who
Friday	15:00	16:00	Blackboard Collaborate	Yan Kin Chi
	Online via Blackboard Collaborate (click Course Room, as described in Lecture 1). Week 2 - 11.			
Monday	14:00	15:00	Blackboard Collaborate	Yan Kin Chi
	Online via Blackboard Collaborate (click Course Room, as described in Lecture 1). Week 2 - 11			
Thursday	19:00	20:00	Blackboard Collaborate	Dominic Wong
	Online via Blackboard Collaborate (click Course Room, as described in Lecture 1). Week 2 - 11.			

12

12

Consultations

Join the consultation using Course Room in Blackboard Collaborate



13

13

WebCMS Forum

- For short questions only (such as clarification of assignment spec or lecture materials)
- Your peers, tutors, or myself can help answer
- For ANY OTHER questions, better see us at the online consultations.
- So we don't expect many questions in the Forum.

14

14

Readings

- No text book
- Slides will be provided / linked from the course homepage
- Core readings (papers) are also provided
- References / supplementary reading list can be found from the course homepage

15

15

Assumed knowledge

Official prerequisite of this course is COMP2521 / COMP1927 / COMP9024.

At the start of this course students should be able to:

- understand bit and byte operations in C/C++.
- write C/C++ code to read from/write to files or memory.
- produce **correct** programs in C/C++, i.e., compilation, running, testing, debugging, etc.
- produce readable code with clear documentation.
- appreciate use of abstraction in computing.

16

16

Learning outcomes

- have a good understanding of the fundamentals of text compression
- be introduced to advanced data compression techniques such as those based on Burrows Wheeler Transform
- have programming experience in Web data compression and optimization
- have a deep understanding of XML and selected XML processing and optimization techniques
- understand the advantages and disadvantages of data compression for Web search
- have a basic understanding of XML distributed query processing
- appreciate the past, present and future of data compression and Web data optimization

17

17

Assessment

```
a1        = mark for assignment 1       (out of 15)
a2        = mark for assignment 2       (out of 35)
asgts     = a1 + a2                     (out of 50)
exam      = mark for final exam         (out of 50)
okEach    = exam > 20                   (after scaling)
mark      = a1 + a2 + exam
grade     = HD|DN|CR|PS  if mark >= 50 && okEach
          = FL           if mark <  50 && okEach
          = UF           if !okEach
```

18

18

One final exam

- One final exam (worth 50 %)
- If you are ill on the day of the exam, do not attend the exam – c.f. fit-to-site policy. Apply for special consideration asap.
- It's likely to be an online exam. More details to be provided later in the course.

19

19

Two assignments

- 1 small prog assignment (15%)
- 1 larger prog assignment (35%)
- Late submission: 5% of the max subtracted from earned marks per day (no acceptance after 5 days late) – *see Course Outline for details.*
- Advanced project in lieu of the assignments is possible. Pre-arrangement with the lecturer before end of week3 if interested.

20

20

Programming assignments

- First assignment is warm-up, relatively easier
- The 2nd assignment is larger in scale, and more challenging
- In addition to correctness, reasonable runtime performance is required
- All submitted code will be checked for plagiarism.

21

21

Tentative assignment schedule

#	Description	Due	Marks
1	Programming assignment 1 (fundamental)	Week 5	15%
2	Programming assignment 2 (compression and search)	Week 9	35%

22

22

In the past, students didn't do well because:

- *Plagiarism*
- Code failed to compile
- Program didn't work in CSE linux
- Late submission
- Program did not follow the spec
- Program failed auto-marking

23

23

Please **do not** enrol if you

- **Don't like the setup of COMP9319 (e.g., no tuts, auto-marking for assigts)**
- Not comfortable with COMP2521 / COMP1927 / COMP9024
- Cannot produce correct C/C++ program on your own
- Have poor time management
- Are too busy to attend lectures

24

24

Tentative course schedule

Week	Lectures	Assignments
1	Introduction, basic information theory, basic compression	
2	More basic compression algorithms	
3	Adaptive Huffman; Overview of BWT	a1 released
4	Pattern matching and regular expression	
5	FM index, backward search, compressed BWT	a1 due; a2 released
6	-	
7	Suffix tree, suffix array, the linear time algorithm	
8	XML overview; XML compression	
9	Graph compression; Distributed Web query processing	a2 due
10	Optional advanced topics; Course Revision	

25

25

Summarised schedule

0. Information Representation (today)
1. Compression
2. Search
3. Compression + Search on plain text
4. “Compression + Search” on Web text
5. Selected advanced topics (if time allows)

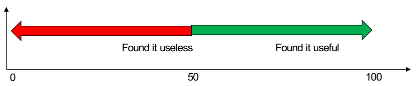
26

26

Is COMP9319 useful ?

It depends on:

1. Your course performance



2. Your field

- Useful in many tech companies / startups such as Google, Amazon
- Not useful for IT applications that system/application performance/scalability is not their priority

27

27

QUESTIONS?

28

Questions to discuss (www)

- What (is data compression)
- Why (data compression)
- Where

29

29

Compression

- Minimize amount of information to be stored / transmitted
- Transform a sequence of characters into a new bit sequence
  - same information content (for lossless)
  - as short as possible

30

30

Familiar tools

- Tools for
  - .Z
  - .zip
  - .gz
  - .bz2
  - …

A glimpse

raaabbccccdabbbbeee$

Run-length coding

- Run-length coding (encoding) is a very widely used and simple compression technique
  - does not assume a memoryless source
  - replace runs of symbols (possibly of length one) with pairs of (symbol, run-length)

RLE

raaabbccccdaaaaabbbbeeeeeed$

ra3bbc4da5b4e6d$

Example: BWT

rabcabcababaabacabcabcabcababaa$

Example: BWT

rabcabcababaabacabcabcabcababaa$

aabbbbccacccrcbaaaaaaaaaabbbbba$

Example: BWT+RLE

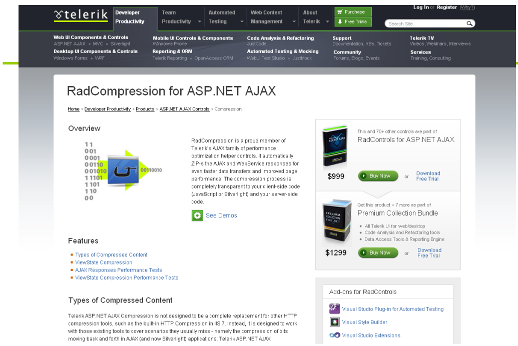
rabcabcababaabacabcabcabcababaa$

aabbbbccacccrcbaaaaaaaaaabbbbba$

aab4ccac3rcba10b5a$

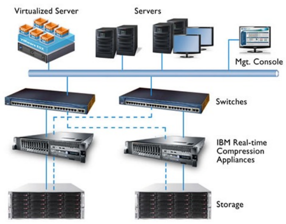
HTTP compression

HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip

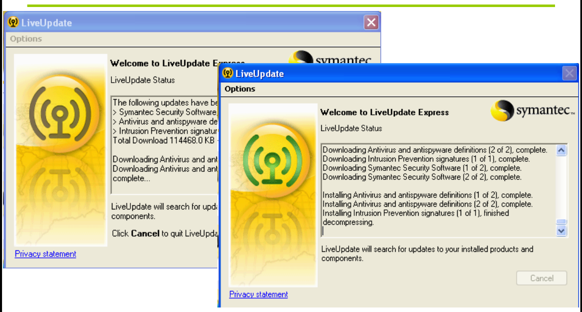


Storwize

Better Storage Utilization
Reduces existing storage utilization up to 80%
No performance degradation
Lowers Capital and Operational Costs
Better Energy Efficiency
Less to store, power and cool
…



Anti-virus definitions & updates



Others

- Software updates
  - e.g., Reg files, UI schemas / definitions
- Software configuration/database updates
  - e.g., Virus database for anti-virus software
- Data streams/Web services
  - e.g., RSS, JSON

Compression & patents

- e.g., STAC vs Microsoft


Microsoft Loses Case On Patent

By LAWRENCE M. FISHER and

A Federal court jury found the Microsoft Corporation guilty of patent infringement today and awarded $120 million in damages to a small California company that had accused Microsoft of appropriating its technology for

- e.g., United States Patent 5,533,051:
  the direct bit encode method of the present invention is effective for reducing an input string by one bit regardless of the bit pattern of the input string.


```
wong:Desktop wong$ ls -l image.jpg
-rwx------@ 1 wong  staff  671172 11 Feb 17:32 image.jpg
wong:Desktop wong$ gzip image.jpg
wong:Desktop wong$ ls -l image.jpg.gz
-rwx------@ 1 wong  staff  424840 11 Feb 17:32 image.jpg.gz
wong:Desktop wong$ mv image.jpg.gz image.jz
wong:Desktop wong$ gzip image.jz
wong:Desktop wong$ ls -l image.jz.gz
-rwx------@ 1 wong  staff  424932 11 Feb 17:32 image.jz.gz
wong:Desktop wong$ mv image.jz.gz image.jzz
wong:Desktop wong$ gzip image.jzz
wong:Desktop wong$ ls -l image.jzz.gz
-rwx------@ 1 wong  staff  425018 11 Feb 17:32 image.jzz.gz
wong:Desktop wong$
```

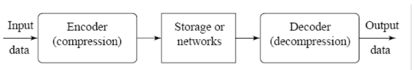

Similarity measure

If two objects compress better together than separately, it means they share common patterns and are similar.

From: Li, M. et al., "The similarity metric", IEEE Transactions on Information Theory, 50(12), 2004

Overview

- Compression refers to a process of coding that will effectively reduce the total number of bits needed to represent certain information.



- Information theory studies efficient coding algorithms
  – complexity, compression, likelihood of error

Compression

- There are two main categories
  – Lossless (Input message = Output message)
  – Lossy (Input message ≠ Output message)
    - Not necessarily reduce quality (example?)

Compression

Compression Ratio = $\frac{\text{Uncompressed Size}}{\text{Compressed Size}}$

Space Savings = $1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}}$

49

49

Example

- Compress a 10MB file to 2MB
- Compression ratio = 5 or 5:1
- Space savings = 0.8 or 80%

50

50

Terminology

- Coding (encoding) maps source messages from alphabet (S) into codewords (C)
- Source message (symbol) is basic unit into which a string is partitioned
  - can be a single letter or a string of letters

51

51

Terminology (Types)

- Block-block
  - source message and codeword: fixed length
  - e.g., ASCII
- Block-variable
  - source message: fixed; codeword: variable
  - e.g., Huffman coding
- Variable-block
  - source message: variable; codeword: fixed
  - e.g., LZW
- Variable-variable
  - source message and codeword: variable
  - e.g., Arithmetic coding

52

52

Terminology (Symmetry)

- Symmetric compression
  - requires same time for encoding and decoding
  - used for live mode applications (teleconference)
- Asymmetric compression
  - performed once when enough time is available
  - decompression performed frequently, must be fast
  - used for retrieval mode applications (e.g., an interactive CD-ROM)

53

53

Decodable

A code is
- distinct if each codeword can be distinguished from every other (mapping is one-to-one)
- uniquely decodable if every codeword is identifiable when immersed in a sequence of codewords

54

54

Example

- A: 1
- B: 10
- C: 11
- D: 101
- To encode ABCD: 11011101
- To decode 11011101: ?

Uniquely decodable

- Uniquely decodable is a prefix free code
  if no codeword is a proper prefix of any other
- For example {1, 100000, 00} is uniquely decodable, but is not a prefix code
  – consider the codeword {…1000000001…}
- In practice, we prefer prefix code (why?)

Example

S	Code
a	00
b	01
c	10
d	110
e	111

Example

S	Code
a	00
b	01
c	10
d	110
e	111

0100010011011000

Example

S	Code
a	00
b	01
c	10
d	110
e	111

0100010011011000

b a b a d d a

Static codes

- Mapping is fixed before transmission
  – E.g., Huffman coding
- probabilities known in advance

Dynamic codes

- Mapping changes over time
  - i.e. adaptive coding
- Attempts to exploit locality of reference
  - periodic, frequent occurrences of messages
  - e.g., dynamic Huffman

61

61

Traditional evaluation criteria

- Algorithm complexity
  - running time
- Amount of compression
  - redundancy
  - compression ratio
- How to measure?

62

62

Measure of information

- Consider symbols s_i and the probability of occurrence of each symbol $p(s_i)$
- In case of fixed-length coding , smallest number of bits per symbol needed is
  - $L \geq \log_2(N)$ bits per symbol
  - Example: Message with 5 symbols need 3 bits ($L \geq \log_2 5$)

63

63

Variable length coding

- Also known as entropy coding
  - The number of bits used to code symbols in the alphabet is variable
  - E.g. Huffman coding, Arithmetic coding

64

64

Entropy

- What is the minimum number of bits per symbol?
- Answer: Shannon's result – theoretical minimum average number of bits per code word is known as Entropy (H)

$$\sum_{i=1}^{n} -p(s_i)\log_2 p(s_i)$$

65

65

Entropy example

- Alphabet S = {A, B}
  - p(A) = 0.4; p(B) = 0.6
- Compute Entropy (H)
  - $-0.4*\log_2 0.4 + -0.6*\log_2 0.6 = .97$ bits
- Maximum uncertainty (gives largest H)
  - occurs when all probabilities are equal

66

66

Example: ASCII

0	nul	1	soh	2	stx	3	etx	4	eot	5	enq	6	ack	7	bel
8	bs	9	ht	10	nl	11	vt	12	np	13	cr	14	so	15	si
16	dle	17	dc1	18	dc2	19	dc3	20	dc4	21	nak	22	syn	23	etb
24	can	25	em	26	sub	27	esc	28	fs	29	gs	30	rs	31	us
32	sp	33	!	34	"	35	#	36	$	37	%	38	&	39	'
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	del

ASCII

- Example: SPACE is 32 or 00100000. ‘z’ is 122 or 01111010
- 256 symbols, assume same probability for each
- P(s) = 1/256
- Optimal length for each char is log 1/P(s) => log 256 = 8 bits

Example 1: 80 days weather

S

R

All sunny days except the last 16 days:

SSS…RRRSSSSSRRRRSSSS

Example 2: 80 days weather

S C

R

All sunny days except the last 16 days:

SSS…RRRCSSSSRRRRCSSS