Course Outline

COMP9319
Web Data Compression and Search
Semester 1 2017

Contents

- Course Staff
- Course Details
- Course Summary
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Staff

Contact Details

Staff Name	Role	Email	Phone
Raymond Wong	Course Convenor	wong at cse.unsw.edu.au	55932

Consultations

Staff Name	Day/Time	Location
Raymond Wong	Wednesday 1100-1200	K17-213

Course Details

Course Code: COMP9319
Course Title: Web Data Compression and Search
Units of Credit: 6
Course WebSite: http://www.cse.unsw.edu.au/~cs9319/
Handbook Entry: http://www.handbook.unsw.edu.au/postgraduate/courses/2017/COMP9319.html

Course Summary

This course will teach you the foundations of:

- how different compression tools work.
- how to manage a large amount of data on small devices.
- how to search gigabytes, terabytes or petabytes of data.
- how to query distributed databases efficiently.
- how to perform full text search efficiently without added indexing.

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., XHTML).

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. The course is composed of the following parts:

- Adaptive coding, information theory
- Text compression (zip, gzip, bzip, etc)

- Burrows-Wheeler Transform and backward search
- XML compression
- Indexing
- Pattern matching and regular expression search
- Distributed querying
- Fast index construction
- Implementation

If time allows, we may cover optional topics such as: streaming algorithms, on-line data analytics, Web data optimization for mobile devices.

The lecture materials will be complemented by three programming assignments and numerous tutorial exercises.

Student Learning Outcomes

Graduate capabilities are the qualities, skills and understandings that the university agrees its students should develop during their studies. Some of these graduate capabilities will be developed and assessed in this course. For example, bonus marks will be given in the optional part of the programming assignments in order to encourage independent learning and critical thinking.

After completing this course, students will:

- 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

There are three programming assignments and one final exam in the course. Written exercises will also be provided and relate to the fundamentals of data compression. The programming assignments will relate to the applications of Web data compression and search techniques and/or their programming.

This course contributes to the development of the following graduate capabilities:

Graduate Capability	Acquired in
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	Programming assignments
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Assignments & classroom discussions
Professionals capable of ethical, self- directed practice and independent lifelong learning	Readings provided in the course
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Case studies

Assumed Knowledge

Official prerequisite of this course is COMP2911/9024.

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

- understand fundamental data structures.
- produce correct programs in C/C++, i.e., compilation, running, testing, etc.
- produce readable code with clear documentation.
- appreciate use of abstraction in computing.

Teaching Rationale

The learning foci in this course are primarily lectures (introduction, illustration, discussion), written exercises (theoretical knowledge) and programming assignments (practical knowledge). The course will have an emphasis on the fundamental techniques and their advantages / disadvantages in the context of real applications. Students will learn the main contents of the course through lectures. Tutorial exercises are available to assist students to obtain in-depth understanding of course materials, and finally the assignments aim to strengthen their understanding and develop practical skills on Web data compression and search.

Teaching Strategies

- Lectures: the main way to introduce concepts, show examples, show/discuss benefits of different techniques.
- Tutorial exercises: assist students to digest the concepts via more discussions and examples (to be discussed in the tutorials or lectures).
- Consultations: weekly consultation sessions to provide personalized advice / assistance to students.

The lectures will serve as the main method of delivering the course contents to the students. Tutorial exercises and programming assignments give students the hands-on experience on various subjects of the course. The consultation will give students the opportunity to seek assistance from the lecturer on all contents of the course.

Assessment

- 3 Programming assignments: 20 points, 30 points and 50 points respectively.
- Final exam: 100 points.
- **Final Mark** : 2 * (Assgt1 + Assgt2 + Assgt3) * FinalExam / (Assgt1 + Assgt2 + Assgt3 + FinalExam)

All programming assignments will be based on the result of auto marking scripts, the quality of the programming code and documentation.

Late programming assignments will attract 10% penalty for the first day and 30% penalty for each subsequent day.

Academic Honesty and Plagiarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism

The pages below describe the policies and procedures in more detail:

- Student Code Policy
- Plagiarism Policy Statement
- Plagiarism Procedure
- Student Misconduct Procedure

Course Schedule

Tentative schedule of the course is as follows:

Week	Lecture
1	Course overview; basic information theory
2	Arithmetic coding, adaptive coding, dictionary coding
3	Adaptive Huffman, LZW; Overview of BWT
4	Pattern matching and regular expression
5	FM index, backward search, suffix array
6	Suffix array O(n), compressed BWT
7	XML overview
8	XML compression
9	Graph compression
10	Distributed Web query processing
11	Optional topics: e.g., streaming data processing
12	Revision

Resources for Students

There will be no textbook used in this course. Lecture slides and supplementary readings will be provided and used.

You may find the readings below useful as reference materials:

- Managing Gigabytes: Compressing and Indexing. Documents and Images, Second Edition. Ian H. Witten,

Alistair Moffat, Timothy C. Bell, Morgan Kaufmann, 1999. (recommended reference, available at the university bookstore)
- Search Engines: Information Retrieval in Practice. W. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education, 2009.
- http://www.data-compression.info contains lots of valuable resources on data compression (especially links to readings and useful advice), despite the website's pink color!
- http://www.data-compression.com contains lots of information, readings, and downloads regarding data compression. It also contains information on image and speech compression, which are beyond the scope of this course.
- Data on the Web: from relations to semistructured data and XML. Serge Abiteboul, Peter Buneman, Dan Suciu. Morgan Kaufmann, 2000.

You will also find your previous textbooks on data structures and/or algorithms useful, in case you need to refer to the fundamentals of data structures and algorithms for text processing.

Last but not least, we will heavily refer to the W3C specs on:

- XML
- XML Information Set
- DOM
- XPath
- XPath Data Model (XDM)
- XQuery

Course Evaluation and Development

This course is evaluated each session using the CATEI system.

The CATEI evaluation from the last time I taught this course showed that students were satisfied with all aspects of the course. Thus we intend to maintain the same style and structure for the up-coming offering. Please note that your feedback is important and will be considered seriously. Student feedback via the CATEI system will enable improvements to future offerings of this subject.