Course Outline

COMP9319
Web Data Compression and Search
Teaching Period 1 2011

Contents

- Course Staff
- Course Details
- 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	Tuesday 1400-1500	J17 434

*Other consultation time is available on Thursday (email appointment is required).

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/2011/COMP9319.html

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 a written assignment, a programming project and numerous tutorial exercises.

Student Learning Outcomes

Graduate attributes are the qualities, skills and understandings that the university agrees its students should develop during their studies. Some of these graduate attributes will be developed and assessed in this course. For example, bonus marks will be given in the programming project 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 one written assignment, one programming project and one final exam in the course. The written assignment will relate to the fundamentals of data compression. The programming project will be related to advanced topics on data compression and Web search.

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 Java, 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 assignment (theoretical knowledge) and project (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. Tutorials are available to assist students to obtain in-depth understanding of course materials and develop practical skills by working on the project.

Teaching Strategies

- Lectures: the main way to introduce concepts, show examples, show/discuss benefits of different techniques.
- Tutorials: assist students to digest the concepts via more examples and tutorial exercises, and to develop skills necessary for completing the programming project.
- Consultations: weekly consultation session 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, written assignment and programming project give the 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

- 1 Written assignment: 20 points.
- 1 Programming project: 80 points.
- Final exam: 100 points.
- **Final Mark** : 2 * (assgt + proj) * FinalExam / (assgt + proj + FinalExam)

The written assignment will be marked according to the quality of the solution as well as the amount of effort demonstrated by the student. The programming project will be based on the result of auto marking scripts, the quality of the programming code and documentation.

Late assignments will receive **Zero** marks.
Late projects will attract 10% penalty for the first day and 30% for each subsequent day.

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Schedule

Schedule of topics, organised by week:

No schedule is available (yet)

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.
- 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.

Your feedback will be considered seriously - especially because this is a new course. Student feedback via the CATEI system will enable improvements to future offerings of this subject.