COMP6714 Course Introduction

Table of Contents

1. Pre-requisites
2. Course philosophy and teaching strategies
3. Course aims
4. Learning outcomes
5. Administrative Components
6. Assessment
7. Reading e-mail
8. Academic honesty and plagiarism
9. Course Material
10. Further information
11. Continual Course Improvement

Units of credit	6
Parallel teaching	no

1. Pre-requisites

The formal pre-requisite for this course is COMP9024 Data Structures and Algorithms.

The knowledge that we assume from COMP9024 is:

- experience with procedural programming, and an understanding of a range of data structures (e.g., trees, graphs, hash-tables) and algorithms (e.g., sorting, divide-and-conquer)

2. Course philosophy and teaching strategies

The learning foci in this course are primarily lectures (theoretical knowledge) and projects (practical knowledge). The course will have an emphasis on problem solving for real applications.

Students will learn the main contents of the course through lectures.

3. Course aims

This course aims to introduce theories and methods of information retrieval, and practical issues regarding search engines.

The course is designed to be practical. As such, real-life examples of information retrieval will also be used throughout the course.

4. Learning outcomes

Students successfully completing this course will be able to:

- understand the whole process of information retrieval and search engines
- understand various document and retrieval models used in information retrieval
- understand various indexing and query processing tehniques and their variants
- develop solutions for real problems using existing technologies
- appreciate the past, present and future of information retrieval and search engine technologies

The learning outcomes are closely related to UNSW graduate attributes 1 — 6. For example, bonus questions will be given in the assignment in order to encourage independent learning and critical thinking.

5. Administrative Components

See the course hompage (http://www.cse.unsw.edu.au/~cs6714) for (up-to-date) information regarding **Course Staff**, **Course Schedule**, and **Course Resource List**.

As the course is continuously developing together with the information retrieval field, the course schedule is subject to change too. Please read the introduction slides (lecture notes of the first week) for the course schedule in the current offering.

6. Assessment

The assessment of the course comprises of written assignments, programming projects and exams.

```
ass1       : 10%
ass2       : 20%
proj1      : 20%
mid-exam   : 20%
final exam : 30%
```

Note that you also need to score at least 40 (out of 100) in the final exam to pass the course.

Both written and programming assignments are helpful to achieve deep understanding of the course materials and develop problem solving abilities.

Relationship to the Learning Outcomes:

- All assignments and the final examination are used to assess whether you have achieved the objectives of the subject.
- The programming assignment, in particular, gives you the opportunities to engage in real problems, and helps you to develop in-depth understanding of the course materials and develop problem-solving skills.

Grading Criteria: Grading criteria for each assessment will be detailed in the specification.

Late submission: Assignments/projects submitted late are subject to late penalties, which are specified in the assignment/project specifications. "Hard" Late penalties are normally used in this course, which reduces the actual mark obtained. Thus if the assignment is marked out of 10, and students *A* and *B* hand in assignments worth 9 and 7, both receiving 20% penalty, *A* gets 9 * (1 - 0.2) = 7.2 and *B* gets 7 * (1 - 0.2) = 5.6.

Assignment submission: Assignment submission procedure is described in the assignment specification document, which will be linked to this page when the assignment specification becomes available. Generally assignments are submitted electronically using the give program running on the School's computer systems (in labs, and on servers). Details are in the assignment specifications.

7. Reading e-mail

You should check your school e-mail frequently in case of announcements relating to this course. We assume that you read e-mail sent to your CSE account by the next working day during teaching sessions.

- How to forward your CSE E-mail to another e-mail address
- CSE E-mail guidelines

8. Academic honesty and plagiarism

Copying assignments is unacceptable. Assignments will be checked. The penalties for copying range from receiving no marks for the assignment, through receiving a mark of 00 FL for the course, to expulsion from UNSW (for repeat offenders). Allowing someone to copy your work counts as plagiarism, even if you can prove that it is your work.

Further details of the School plagiarism policy can be found here. (You acknowledged receipt of these rules when you obtained your CSE computer account, and the link above is for your convenience so that you can review the rules now.)

We are aware that a lot of learning takes place in student conversations, and don't wish to discourage those. However, it is important, for both those helping others and those being helped, not to provide/accept any programming language code in writing, as this is apt to be used exactly as is, and lead to plagiarism penalties for both the supplier and the copier of the codes. Write something on a piece of paper, by all means, but tear it up/take it away when the discussion is over.

If you are new to studying in Australia, be aware that attitudes to plagiarism at UNSW may be different from those in your home country. Make sure you are clear about the rules here at UNSW. In brief, and for the purposes of COMP9318, plagiarism includes copying or obtaining all, or a substantial part, of the material for your assignment, whether programming language code, or written or graphical report material, without written acknowledgement in your assignment from:

- location on the internet;
- a book, article or other written document (whether published or unpublished) whether electronic or on paper or other medium;
- another student, whether in your class or another class;
- a non-student (e.g. from someone who writes assignments for money)

Note that if you copy code or other material from another student or non-student with acknowledgement, you will not be penalised for plagiarism, but you are unlikely to get any marks for the copied material. If you use code found in a publication (on the internet or otherwise) then the marks you get for this will be at the

marker's discretion, and will reflect the marker's perception of the amount of work you put into finding and/or adapting the code, and the degree to which you understand the code.

Note also that there is a big difference between being able to understand someone else's code, and writing that code yourself from scratch. A computer programmer has to be able to write code from scratch. The assignments provide opportunities for you to develop the skills necessary to write your own code. Use these opportunities!

9. Course Material

- All lecture notes, tutorial and lab notes will be posted on the course web site. These notes summarises the major contents and help you to understand the materials when you read the textbook later. You definitely need to read the corresponding chapters in the textbook to gain a full understanding of the course materials.

```
[MRS08] Christopher D. Manning, Prabhakar Raghavan and Hinrich
Schutze, Introduction to Information Retrieval, Cambridge
University Press. 2008.
```

```
[CMS09] W. Bruce Croft, Donald Metzler, and Trevor Strohman,
Search Engines: Information Retrieval in Practice. Pearson. 2009.
```

- Reference books for this course are:

```
[BB99] Ricardo Baeza-Yates and Ber thier Ribeiro-Neto, Modern Information
Retrieval, Addison Wesley. 1999.
```

- Useful Resources
  - http://nlp.stanford.edu/IR-book/information-retrieval.html

10. Further information

- Students enrolled in the course are expected to attend all classes
- The use of School of Computer Science and Engineering computing laboratories is subject to rules described in the Yellow Form, which you acknowledge (electronic) receipt of when you receive your computing account. The Yellow Form also outlines what to do in case illness or misadventure that affects your assessment, and supplementary examinations procedures within the School of Computer Science and Engineering.
- Information on UNSW Occupational Health and Safety policies and expectations
- Equity and Diversity issues

11. Continual Course Improvement

Each year feedback is sought from students and other stakeholders about the courses offered in the School and continual improvements are made based on this feedback. UNSW's Course and Teaching Evaluation and Improvement (CATEI) Process (http://www.unsw.edu.au/learning/pve/catei.html) is one of the ways in which student evaluative feedback is gathered. Significant changes to courses and programs within the School are communicated to subsequent cohorts of students.

Wei Wang, June 2011

Last updated 2010-07-20 22:30:15 EST