COMP6714: Information Retrieval and Web Search

Introduction
Outline

• Introduction of the course
• Introduction of IR and Web Search
Lecture in Charge

•Lecturer-in-charge:
  •Dr. Yifang Sun
  •Email: yifangs@cse.unsw.edu.au
    • use [comp6714] in subject (otherwise I may miss your email)
    • I will reply your email in 24 hours on weekdays and 48 hours on weekends
Lectures

- Online lecture with pre-recorded slides
  - location: anywhere you like
  - Software: MS Teams
    - Invitation will be sent to you every Wed and Fri morning
- Time (Sydney time)
  - Wed 1300 - 1500
  - Fri 1100 - 1300
  - Recording can be found in MS Teams

- Slides on course website
  - http://www.cse.unsw.edu.au/~cs6714

- QA sessions
  - You can ask during the lectures (and I will pause the video and answer)
  - Ask in the forum (i.e., piazza) or during online consultations
  - Will address common questions in forum at the beginning of each lecture

- Schedule and length of lectures may vary based on the progress of the course

- Note: attending/watching every lecture is assumed
Consultations

• Online QA discussions using Piazza
  • https://piazza.com/class/ksfa3ry3ud5md
  • encourage every student to participate
  • Raise questions and try to help others

• Online consultation
  • 1300 – 1400 every Friday
  • using MS Teams

• Private online consultation with LiC
  • please book an appointment with me with a brief description of your questions, with [comp6714] in subject
  • only for problems that cannot be solved in the forum and during the online consultation
Course Aims

• Not possible to cover every aspect of Information Retrieval and Web Search

• We will focus on
  • concepts
  • algorithms
  • principles

• We will not focus on
  • programming languages and API
  • specific platforms/tools

• Make use of tutorials and documents on the Internet
Expectation

• What are expected in this course
  • Many modules covering a **broad spectrum** of IR/NLP/SE
  • Heavy workload expected: must read and digest the **textbook and slides + additional notes**
  • Requires substantial **algorithm/data structure** design/analysis experience & capability + some maths.
  • **Up-to-date** viewpoints, understanding, knowledge (from the academia & industry)
  • ➔ Plan your time well

• You are welcome to ask questions at anytime
• Review after the lecture
Knowledge Assumed (non-exhaustive)

• Data structures & algorithms:
  • Heap/priority queue: build a heap in O(n) time?
  • Membership query: tradeoffs? worst/avg-case time complexities = ?
• Recursion:
• DFS/BFS/Best-first search

Given an array A of integers. Design an algorithm to return two elements x, y in A, such that x + y = 100 if any, and
1. the algorithm takes O(n*\log(n)) time, or
2. the algorithm takes O(n) time
Knowledge Assumed (non-exhaustive)/2

• C/C++ & Python Programming:
  • Pointer
  • sizeof(int) = ? sizeof(p) = ? sizeof(*p) = ?
    sizeof(str) = ?
  • Be able to learn to use new Python libraries and write & debug python programs
    • Quickly learn a python-based framework in this course
Knowledge Assumed (non-exhaustive)/3

• CS Architecture
  • Memory hierarchy: name the levels?
  • Bit representation: binary string for any x? How to obtain the 3rd-5th bits of a byte?

• Maths
  • Calculus: How to find the minimum/minimal value of a function f(x)?
  • Probabilities and statistics: rv; linearity of expectation; indicator variable; number of heads by tossing a biased coin n times; Bayesian theorem
  • Linear algebra: inner/dot product of u and v = ? matrix multiplication
Assessment

• One written assignment (25%)
• One programming project (25%)
• Final exam (50%)

• Most likely to due after week 7
• Details to be available later
Written Assignment

• Exam-style questions
  • computational, short answer
  • no essay, no multiple choice

• Regarding the lecture contents
  • algorithms, principles, …
  • to assess your understanding, not memory

• Late penalty
  • firm deadline
  • zero mark for late submission
Programming project

• Individual task
• Both results and source codes will be checked.
  • Zero mark if your codes cannot be run due to some bugs.
• Late penalty
  • 10% reduction of raw marks for the 1st day, 30% reduction per day for the following 3 days
Final exam

• Open book exam
• Firm deadline
• No supplementary exam will be given if you fail
• Special consideration must be submitted prior to the start of the exam
Warning

• This course has
  • Broad coverage
  • Heavy workload
  • High fail rate ≥ 20%

• Specially, we do not accept personal plea or excuses
  • if you have valid reasons that affect your performance, apply for a UNSW Special Consideration
Warning - cont.

• Common excuses/arguments
  • I spent so much time and effort on this course but still failed?
  • I did the work by myself and may have shared it with my classmate for discussion.
  • If I fail this course, I will [...]. Please.
Academic honesty and plagiarism

- Zero tolerance to plagiarism
  - You will get 0 marks
- Examples of misconduct:
  - Copy other students’ work
  - Let other students copy your work
  - Copy from GitHub
  - Find a ghost writer
  - …
- I will not accept the following excuses:
  - “I’ve left the lab with my screen unlocked”
  - “He stole it from my computer”
  - “I only gave my code to A. A didn’t use it but gave it to B”
  - …
Please do not enrol if you…

• Don’t have the required knowledge
• Cannot produce correct Python program on your own
• Have poor time management
• Are too busy to watch lecture videos
• Are not honest

• Otherwise, you are likely to perform badly in this subject
# Tentative course schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Labs/Assignment/Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Course Introduction + Boolean Retrieval</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Preprocessing</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Index construction</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Compression</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Vector Space Model</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Flexibility Week (no lecture)</td>
<td>project</td>
</tr>
<tr>
<td>7</td>
<td>Evaluation</td>
<td>Assignment</td>
</tr>
<tr>
<td>8</td>
<td>Crawling</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Link Analysis</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Revision and Exam Preparation</td>
<td></td>
</tr>
</tbody>
</table>
General Suggestions

• Make use of LiC and tutors
  • don’t hesitate to ask questions

• Make use of the forum
  • read the notices in course website and Piazza
  • participate in the discussions in Piazza

• Make use of course materials
  • understand lecture slides
  • read specifications carefully

• Do not misconduct
Your Feedbacks are Always Welcome

• Please advice where I can improve after each lecture, through Piazza or by email

• myExperience system
Outline

• Introduction of the course

• Introduction of IR and Web Search
What is Information Retrieval?

• Let’s start with **Search**

• Search on the Web is a daily activity for many people throughout the world

• Search and communication are most popular uses of the computer

• Applications involving search are everywhere

• The field of computer science that is most involved with R&D for search is *information retrieval* (IR)
Information Retrieval

• Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

• These days we frequently think first of web search, but there are many other cases:
  • E-mail search
  • Searching your laptop
  • Corporate knowledge bases
  • Legal information retrieval

• Primary focus of IR since the 50s has been on text and documents
Search Engine

• As a user
  • What you can do?
  • What to expect?

• As a server
  • How to meet the users’ requirements?
  • How to improve the users’ experience?

• As an observer
  • How to evaluate a search engine?
Basic assumptions of Information Retrieval

• **Collection**: A set of documents
  • Assume it is a static collection for the moment

• **Goal**: Retrieve documents with information that is relevant to the user’s information need and helps the user complete a task
What is a Document?

• Examples:
  • web pages, email, books, news stories, scholarly papers, text messages, Word™, Powerpoint™, PDF, forum postings, patents, IM sessions, etc.

• Common properties
  • Significant text content
  • Some structure (e.g., title, author, date for papers; subject, sender, destination for email)
Unstructured (text) vs. structured (database) data in the mid-nineties
Unstructured (text) vs. structured (database) data in 2019
Documents vs. Database Records

• Database records (or *tuples* in relational databases) are typically made up of well-defined fields (or *attributes*)
  • e.g., bank records with account numbers, balances, names, addresses, social security numbers, dates of birth, etc.

• Easy to compare fields with well-defined semantics to queries in order to find matches

• Text is more difficult
Documents vs. Records

• Example bank database query
  • Find records with balance > $50,000 in branches located in Amherst, MA.
  • Matches easily found by comparison with field values of records

• Example search engine query
  • bank scandals in western mass
  • This text must be compared to the text of entire news stories
Comparing Text

• Comparing the query text to the document text and determining what is a good match is the core issue of information retrieval.

• Exact matching of words is not enough.
  • Many different ways to write the same thing in a “natural language” like English.
  • e.g., does a news story containing the text “bank director in Amherst steals funds” match the query?
  • Some stories will be better matches than others.
Dimensions of IR

• IR is more than just text, and more than just web search
  • although these are central
• People doing IR work with different media, different types of search applications, and different tasks
Other Media

• New applications increasingly involve new media
  • e.g., video, photos, music, speech
• Like text, content is difficult to describe and compare
  • text may be used to represent them (e.g. tags)
• IR approaches to search and evaluation are appropriate
## Dimensions of IR

<table>
<thead>
<tr>
<th>Content</th>
<th>Applications</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Web search</td>
<td>Ad hoc search</td>
</tr>
<tr>
<td>Images</td>
<td>Vertical search</td>
<td>Filtering</td>
</tr>
<tr>
<td>Video</td>
<td>Enterprise search</td>
<td>Classification</td>
</tr>
<tr>
<td>Scanned docs</td>
<td>Desktop search</td>
<td>Question answering</td>
</tr>
<tr>
<td>Audio</td>
<td>Forum search</td>
<td></td>
</tr>
<tr>
<td>Music</td>
<td>P2P search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Literature search</td>
<td></td>
</tr>
</tbody>
</table>
IR Tasks

• Ad-hoc search
  • Find relevant documents for an arbitrary text query
• Filtering (aka information dissemination)
  • Identify relevant user profiles for a new document
• Classification
  • Identify relevant labels for documents
• Question answering
  • Give a specific answer to a question
Big Issues in IR

• Relevance
  • What is it?
  • Simple (and simplistic) definition: A relevant document contains the information that a person was looking for when they submitted a query to the search engine
  • Many factors influence a person’s decision about what is relevant: e.g., task, context, novelty, style
  • *Topical relevance* (same topic) vs. *user relevance* (everything else)
Big Issues in IR

• Relevance
  • Retrieval models define a view of relevance
  • Ranking algorithms used in search engines are based on retrieval models
  • Most models describe statistical properties of text rather than linguistic
    • i.e. counting simple text features such as words instead of parsing and analyzing the sentences
    • Statistical approach to text processing started with Luhn in the 50s
    • Linguistic features can be part of a statistical model
Big Issues in IR

• Evaluation
  • Experimental procedures and measures for comparing system output with user expectations
    • Originated in Cranfield experiments in the 60s
  • IR evaluation methods now used in many fields
  • Typically use test collection of documents, queries, and relevance judgments
    • Most commonly used are TREC collections
  • Recall and precision are two examples of effectiveness measures
Web Search

• New Challenges

• How to obtain data?

• Additional features for Web data