

# Advanced topics in Computer Science

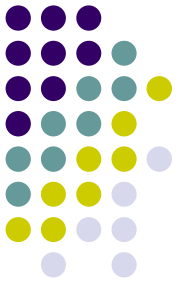
Jiaheng Lu

**Department of Computer Science**

**Renmin University of China**

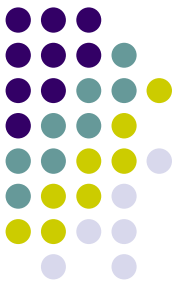
[www.jiahenglu.net](http://www.jiahenglu.net)

# Course purpose



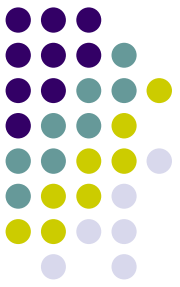
- **Teach in English in most time**
- **Introduce senior undergraduate students to some advanced topics in computer science**

# Course contents



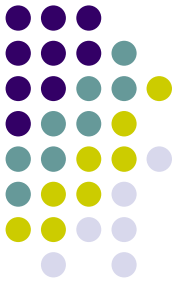
- **Introduction to information retrieval**
- **Approximate string processing**
- **XML data management**
- **Cloud computing**

# Course contents



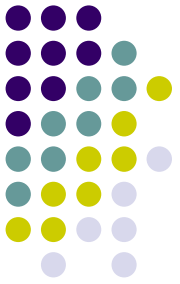
- **Introduction to information retrieval**
- **信息检索和搜索引擎技术**
  - **Basic indexing and tokenization**
  - **“Tolerant” retrieval**
  - **Index construction**
  - **Dictionary and Postings compression**

# Course contents



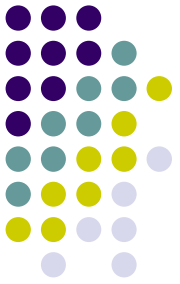
- **XML data management**
- **XML数据管理**
  - XML, XPath, XQuery
  - XSLT, XML Schema
  - XML query processing
  - XML database

# Course contents



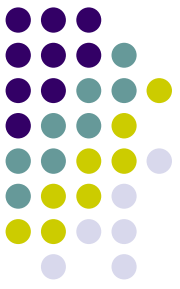
- **String processing and matching**
- 字符串处理技术
- **Exact string matching and approximate string matching**

# Course contents

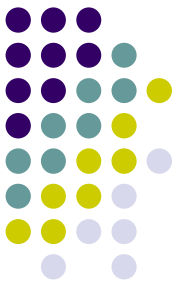


- **Cloud computing**
- 云计算技术
  - **Introduction to cloud computing**
  - **Cloud-based service**
  - **Cloud-based data management**

# Course grading



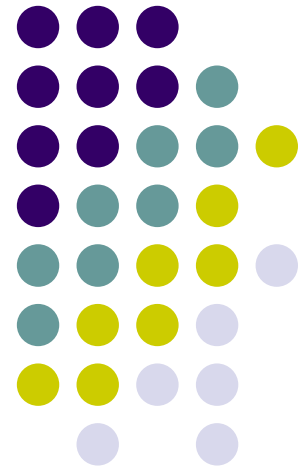
- **Presentation in English/Chinese only 40%**
- **Paper in English only 40%**
- **In-class presence and quiz 20%**

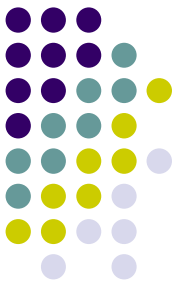


**Any question and any comments ?**

# Evaluating search engines

---

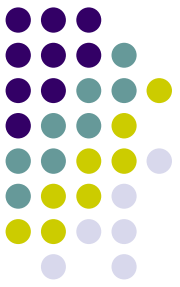




# search engine

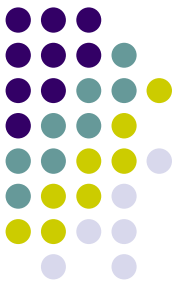
- Have you any comments about search engine?
- Baidu
- Google
- Sogou
- Yahoo

# Measures for a search engine

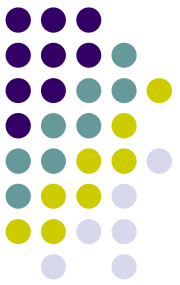


- How fast does it index
  - Number of documents/hour
  - (Average document size)
- **How fast does it search**
  - **Latency as a function of index size**
- Expressiveness of query language
  - Speed on complex queries

# Measures for a search engine

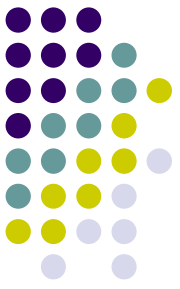


- All of the preceding criteria are *measurable*: we can quantify speed/size; we can make expressiveness precise
- **The key measure: user happiness**
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness



# Measuring user happiness

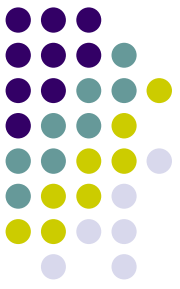
- Issue: who is the user we are trying to make happy?
  - Depends on the setting
- Web engine: user finds what they want and return to the engine
  - Can measure rate of return users
- eCommerce site: user finds what they want and make a purchase
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?



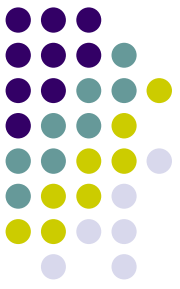
# Measuring user happiness

- Enterprise (company/govt/academic): Care about “user productivity”
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access ... more later

# Happiness: elusive to measure



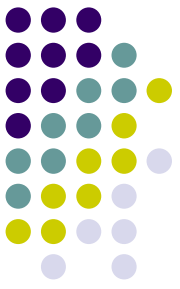
- But how do you measure relevance?
  - Will detail a methodology here, then examine its issues
- Requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A binary assessment of either Relevant or Irrelevant for each query-doc pair



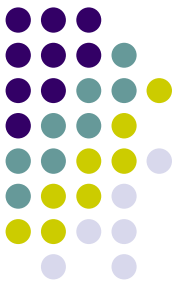
# Evaluating an IR system

- Note: **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- Query: **wine red white heart attack effective**

# Standard relevance benchmarks



- TREC - National Institute of Standards and Testing (NIST) has run large IR benchmark for many years
- Reuters and other benchmark doc collections used
- “Retrieval tasks” specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Irrelevant
  - or at least for subset of docs that some system returned for that query



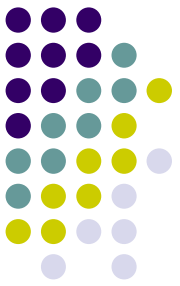
# Precision and Recall

- **Precision:** fraction of retrieved docs that are relevant =  $P(\text{relevant}|\text{retrieved})$
- **Recall:** fraction of relevant docs that are retrieved =  $P(\text{retrieved}|\text{relevant})$

	Relevant	Not Relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

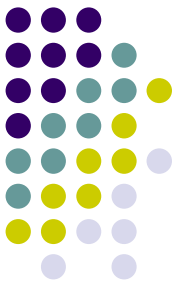
- Precision  $P = \text{tp}/(\text{tp} + \text{fp})$
- Recall  $R = \text{tp}/(\text{tp} + \text{fn})$

# Accuracy – a different measure

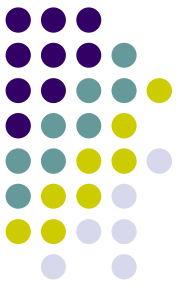


- Given a query an engine classifies each doc as “Relevant” or “Irrelevant”.
- Accuracy of an engine: the fraction of these classifications that is correct.

# Why not just use accuracy?



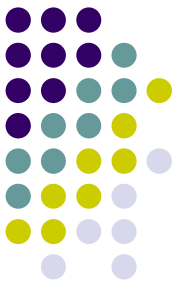
- How to build a 99.9999% accurate search engine on a low budget....
- People doing information retrieval want to find *something* and have a certain tolerance for junk.



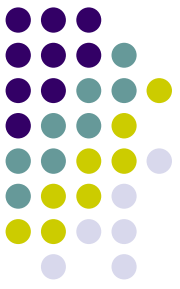
# Precision/Recall

- Can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
  - Precision usually decreases (in a good system)

# Difficulties in using precision/recall



- Should average over large corpus/query ensembles
- Need human relevance assessments
  - People aren't reliable assessors
- Assessments have to be binary
  - Nuanced assessments?
- Heavily skewed by corpus/authorship
  - Results may not translate from one domain to another

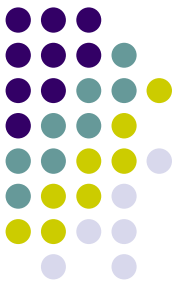


# A combined measure: $F$

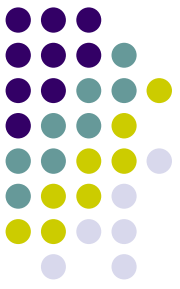
- Combined measure that assesses this tradeoff is  $F$  measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced  $F_1$  measure
  - i.e., with  $\beta = 1$  or  $\alpha = \frac{1}{2}$



**Any question and any comments ?**

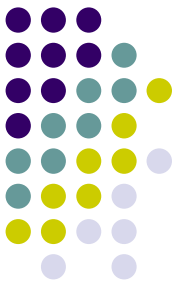


# Precision and Recall

- **Precision:** fraction of retrieved docs that are relevant =  $P(\text{relevant}|\text{retrieved})$
- **Recall:** fraction of relevant docs that are retrieved =  $P(\text{retrieved}|\text{relevant})$

	Relevant	Not Relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision  $P = \text{tp}/(\text{tp} + \text{fp})$
- Recall  $R = \text{tp}/(\text{tp} + \text{fn})$



# Precision and Recall Quiz

	Relevant	Not Relevant
Retrieved	10	3
Not Retrieved	5	2

- Precision  $P = tp/(tp + fp) = 10/13 = 77\%$
- Recall  $R = tp/(tp + fn) = 10/15 = 67\%$