

# Information Retrieval and Extraction

11.8.2020

## Course Logistics

### 3 major parts:

- 4 sessions: introduction from multiple aspects
- 9? " : IR fundamentals:
  - └ models
  - scoring functions
  - crawling, evaluation, etc.
- NLP required specifically for IR
- ML for IR
- : IE fundamentals
  - majorly: NER
- : Information Access / Applications of IR
  - mining specifics: social media, sentiments,
  - computational advertising
  - sentiment analysis.

### Tutorials: Thu/Tue evening

### Grading:

Quizzes / In-class:	10%
Assignments (best 3/4)	15%
Project	60%
└ mini	20
└ major	40
Term Paper	15%
(Final Exam, can be done anytime)	

X \_\_\_\_\_ X \_\_\_\_\_

### Recommended Books

- Stanford IR book
- Search Engines in Practice

### Project Details

#### MINI

- Individual
- 4 weeks, starting today
- Deliverables:
  - 1.
  - 2.

#### Long:

Py / C++ / Java

### DEADLINES

1. 24th August: offline
2. 7th Sept: online + offline

Design and develop a scalable and efficient search engine on Wikipedia.

### REQUIREMENTS

- Query <sup>1s</sup> → result
- Support "Field Queries"<sup>11</sup>
- Total index size: < 1/4th the size of the doc repository

- build your own indexing scheme

### EVAL

- |         |   |                     |
|---------|---|---------------------|
| online  | { | • Search time       |
|         |   | • search efficiency |
| offline | { | • Indexing time     |
|         |   | • Indexing size     |

1. Field Query: Searching within a subset.

26th: Dummy Queries

## MAJOR

- Team of 4 constrained choice
- 10 weeks.

## Advanced topics

Scope well defined (by us)

## 5 touchpoints

Report to mentor every 2 weeks

## 3 Evaluations

first deliverable: scope doc  
(26<sup>th</sup> Sept)

MVP deliverable: full system, v1  
(25<sup>th</sup> Oct)

Complete System: Demo, presentation report  
(14<sup>th</sup> Nov) code, etc.

## Basic Overview

- Searching is important
- Computational advertising

13. 8.2020

R  
E  
C  
A  
P

- Information Retrieval: Science of finding documents of unstructured information
- Economic viability of IR
- Standard Search Engines, and the new players (Fb, Amazon, etc)

## History:

- 1993:
  - new - WWW
  - no search engines
  - new URLs added manually to CERN "what's new"

[Aside] - 1990:

- Archie. (Archie - v), First "search engine"
- U. Minnesota: → Veronica → Jughead.

- First .robo search program www Automated (developed at MIT)
- Nov. 1993. 2<sup>nd</sup> websearch engine. (Aliweb)

- 1994:
  - First proper webcrawler (also looks at content).
  - Another: Lycos, in CMU
  - Many others.

- 1995:
  - Two prominent technologies:

- crawling
- indexing

## 1. CRAWLING:

- problems:

- 1a) Need diversity in seed URLs
- b) Need diversity in crawling strategy.

2. Pages have a lifecycle.

- TOI, Twitter change every minute
- Something like IIT changes occasionally.
- Some websites never change.

→ strat: estimate lifecycles.

1. After 1 crawl, crawl again.
  - if page has changed, increase freq. for it.
  - if page has not changed, decrease freq. for it.

## 2. INDEXING

- helps find URL and content.
- Many classes of indexing (schemes) exist.

Improvements to be made: coverage, scale

- Yahoo starts from Stanford in 1994.
  - Introduces the idea of "browse", searching enabled within a "topic directory".
- the web started growing faster than human editors could keep up
  - introduction of classification and clustering programs

### • 1998: Google → PageRank

- Even if you know content of each page, that is not enough.
- Value assigned by reputation of a given page.
- Need to rank pages for query by reputation.
- think of the web as a massive graph, where each URL has inlinks, and outlinks.
- Rank: the more inlinks there are, recursively.

- domain specific search engines: let you incorporate more domain knowledge than generic search can do.

IR Cycle

Documents  
Containing your  
information  
could be WWW  
or enterprise DB

Source Selection

Can be changed.

Query Formulation

Queries are processed and refined before being sent. Can use your context to improve the query. Like location, or in the direction you are driving.

Processing stuff like synonyms, NER, NLP for IR (chunk, pos, etc)

Search

Tries to prioritize which results are important. Ranking, for example.

Selection

Examining behaviour after first search to modify subsequent behaviour

ExaminationDelivery

- Tries to improve the search experience for the user.
- All of this happens while the user is typing.

- About:
  - quality
  - speed
  - variety of documents

Central problem with search

- As a searcher, I have information need in my head, expressed as a few terms. } Diff b/w concepts query terms
- As a document writer, I also have a concept in my head expressed as a few document terms. } Another diff.

Both are suboptimal

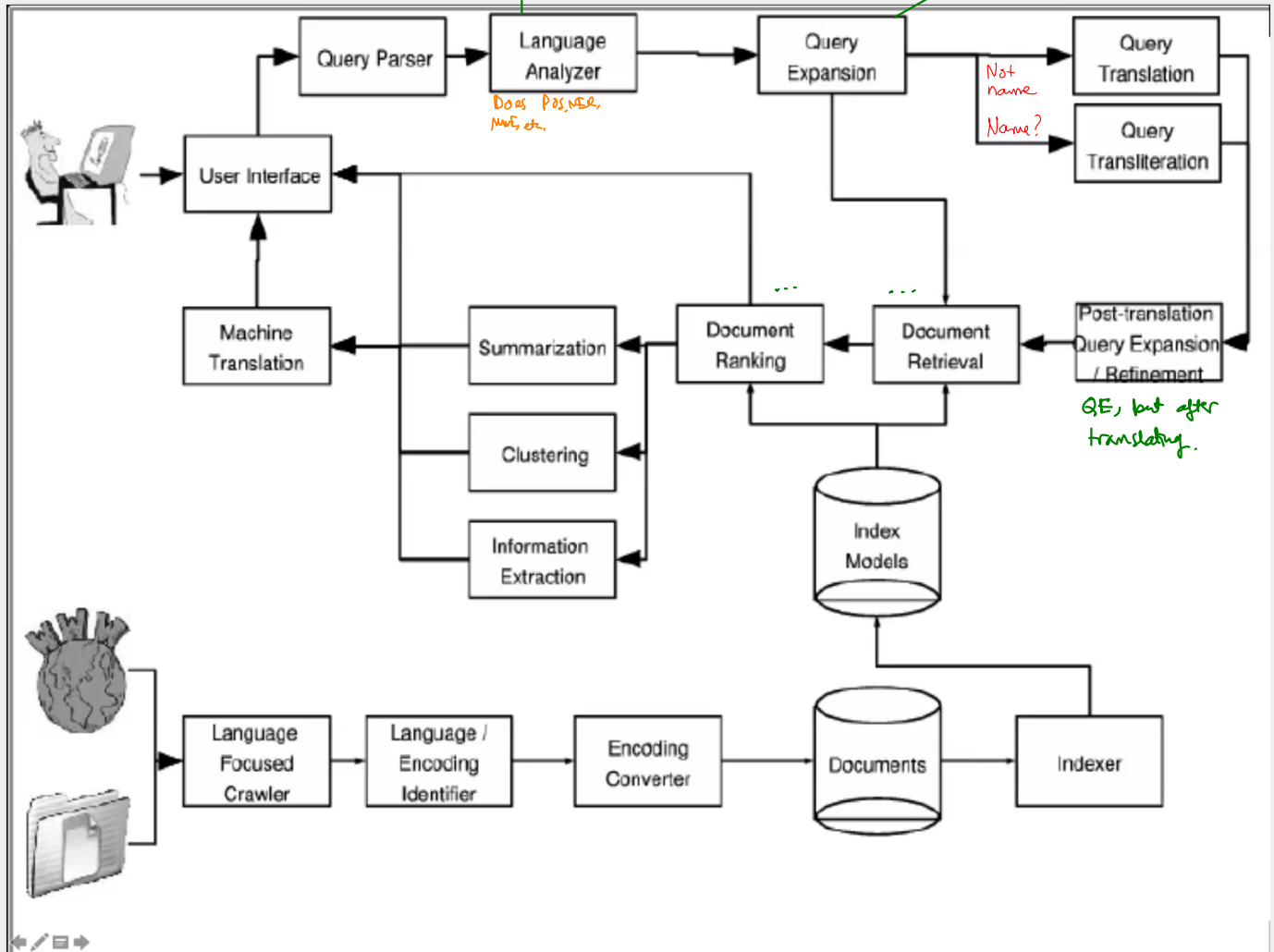
- Job of search engine: Match the (suboptimal) Document Terms and Query Terms

Map the meaning, not the words  
hence, search is complex.

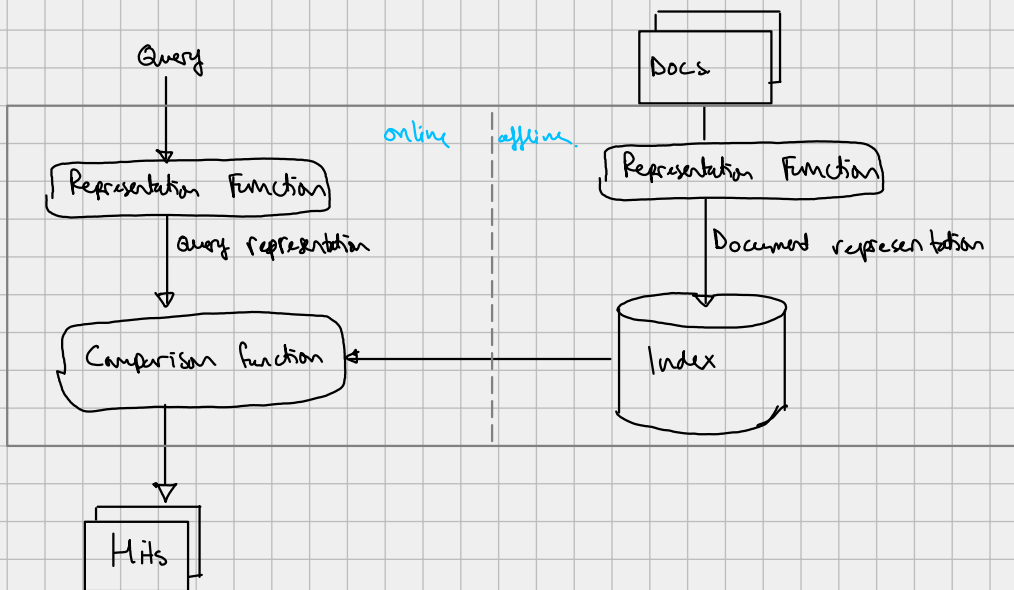
## Cross-Language IR system by IRE

- Auto identification of lang (when not unique) hard  
 ↑ - language ≠ script

Get: synonyms, spellcheck, etc.



## Abstract IR arch



20.3.20

• How do we approach retrieving data

1. Looking at the content
2. Looking at the meta data

Labeling by processing metadata

- But this is not a replacement
- No IR without content.
  - But metadata helps.

Document-query similarities

Much of IR depends on one idea:

- Documents are similar to each other if they are relevant to the same query.
- Likewise

• How is similarity measured?

- $\{\text{query vocab}\} \subset \{\text{doc. vocab}\}$
- prob model based on term occurrences
- Bag of words.

- How to compare search engines? hard.  
a given query will determine quality.

- Getting unstructured data hard.