IR Models: The Boolean Model

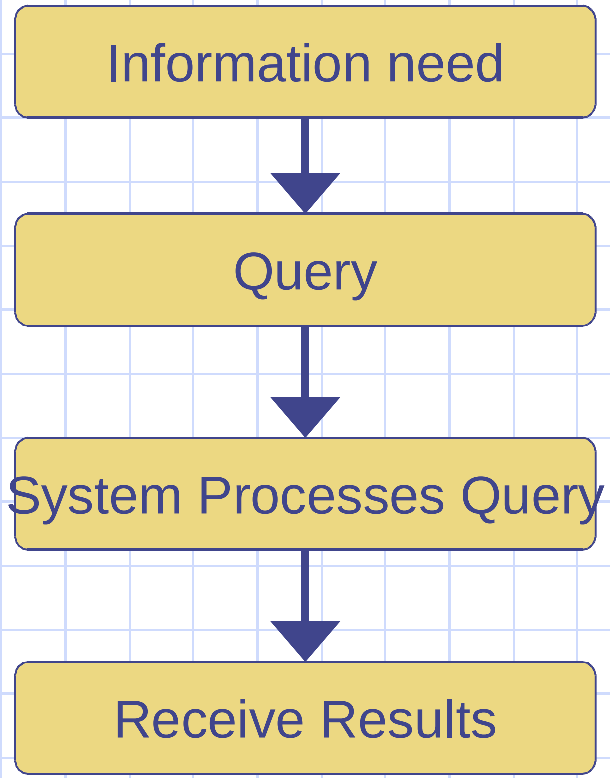
Lecture 6

Where are we now?

- Text Processing
- Inverted Index construction
  - data structures, algorithms, compression…
  - a set of scalable, efficient data structures for finding words in large text collections
- Now, let's take it back to the *problem:* Information Retrieval

Search: Part of a user task

- Classical IR
  - This is central task
  - Add feedback loop where user refines query
- Modern IR
  - Part of the Big Picture
  - An essential tool
  - Used in search, filtering, and browsing



Handling User Queries

- Goal of the search component
  - predict which documents are *relevant* to the user's need
  - *rank* the documents in order of predicted likelihood of relevance to the user.
- Need a model which encompasses
  - documents
  - queries
  - ranking functions

Information Retrieval Models

- A retrieval model consists of:

  D: representation for documents

  R: representation for queries

  F: a modeling framework for D, Q, and the relationships among them

  R(q, d_i): a ranking or similarity function which orders the documents with respect to a query.

Classical IR Models

- Boolean model
- Vector Space model
- Probabilistic model

The Boolean Model

- Based on set theory and Boolean algebra
  - Documents are sets of terms
  - Queries are Boolean expressions on terms
- Historically the most common model
  - Library OPACs
  - Dialog system
  - Many web search engines, too

The Boolean Model, Formally

D: set of words (indexing terms) present in a document

- each term is either present (1) or absent (0)

Q: A boolean expression

- terms are index terms
- operators are AND, OR, and NOT

F: Boolean algebra over sets of terms and sets of documents

Boolean Relevance Prediction

R: a document is predicted as relevant to a query expression iff it *satisfies* the query expression

$$((\mathit{text} \vee \mathit{information}) \wedge \mathit{retrieval} \wedge \neg \mathit{theory})$$

- Each query term specifies a set of documents containing the term
- AND ($\wedge$): the intersection of two sets
- OR ($\vee$): the union of two sets
- NOT ($\neg$): set inverse, or really set difference

Boolean Relevance example

$$((text \vee information) \wedge retrieval \wedge \neg theory)$$

- *“Information Retrieval”*
- *“Information Theory”*
- *“Modern Information Retrieval: Theory and Practice”*
- *“Text Compression”*

Implementing the Boolean Model

- First, consider purely *conjunctive* queries

$$(t_a \wedge t_b \wedge t_c)$$

- Only satisfied by a document containing all three terms
- If $D(t_a) = \{ d \mid t_a \in d \}$, then
  - the *maximum possible* size of the retrieved set is the size of the smallest $D(t_a)$
  - $|D(t_a)|$ is the length of the inverted list for t_a

Algorithm for AND queries

1. For each query term t
   1. retrieve lexicon entry for t
   2. note f_t and address of I_t (inverted list)
2. Sort query terms by increasing f_t
3. Initialize *candidate set* C with I_t of the term with the smallest f_t
4. For each remaining t
   1. Read I_t
   2. For each $d \in C$, if $d \notin I_t$, C <- C – {d}
   3. If C = {}, return… there are no relevant docs
5. Look up each $d \in C$ and return to the user

“Eating cold porridge”

(eat AND cold AND porridge)

1. Pease porridge hot,
2. pease porridge cold,
3. Pease porridge in the pot,
4. Nine days old.
5. Some like it hot,
6. some like it cold,
7. Some like it in the pot,
8. Nine days old.

Beyond AND

- Consider a query that is a conjunction of disjunctions (AND of OR’s)

  (text OR data OR image) AND
  (compression OR compaction) AND
  (retrieval OR indexing OR archiving)

- Treat each disjunction as a single term
  - merge the inverted lists for each OR’d term
  - or, just add the f_t values for a worst-case approximation of the candidate set size

Thoughts on the Boolean Model

- Very simple model based on sets
  - easy to understand and implement
- Only retrieves *exact* matches
  - No ranking of documents: r() is boolean
  - Retrieves too much, or too little
- Sets are easy, but complex Boolean expressions aren't
  - 'cats and dogs' vs. (cats AND dogs)
- All terms are equally important