# CMPS 561
# Boolean Retrieval

Ryan Benton

Sept. 7, 2011

# Algorithms for Intersection

# Algorithms – Basic Intersection (aka Merging)

- Intersect(p1, p2)
  - answer ← {}
  - While (p1 != NIL) and (p2 != NIL) Do
    - if docID(p1) = docID(p2)
      - Then ADD(answer, docID(p1))
        - » p1 ← next(p1)
        - » p2 ← next(p2)
      - Else if (docID(p1) < docID(p2))
        - » Then p1 ←next(p1)
        - » Else p2 ← next(p2)
  - Return answer

# Algorithms – Intersection

- Complexity: $O(x + y)$
  - For any given two posting lists
    - List A has size x
    - List B has size y
  - Note, this is upper bound.
- Formally, Complexity: $\Theta(N)$
  - N can be either
    - Number of documents in collection
  - Note, this is a tight bound.

# Observation

- In many cases, Boolean queries
  - Conjunctive in nature
- Allows for a possible improvement based on posting size (term frequency)

# Algorithms – Conjunctive Query Merging

- IntersectConjunct($t_1$, $t_2$, …, $t_z$)
  - Terms ← SortByIncreasingFrequency(($t_1$, $t_2$, …, $t_z$))
  - Results ← postings(first(Terms))
  - Terms ← rest(Terms)
  - while (Terms != NIL) and (Results != NIL) Do
    - Results ← Intersect(result, postings(first(Terms)))
    - Terms ← rest(Terms)
  - Return Results

# Why?

- By using least frequent term
  - All results guaranteed to be no larger than least frequent term
- In practice
  - The ‘intermediate’ list always places upper bounds on the size.

# References


- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, **Introduction to Information Retrieval**, Chapter 1, 2008.
- Abraham Bookstein and William Cooper, "A General Mathematical Model for Information Retrieval Systems", The Library Quarterly, Vol 26, no. 2, pp 153-67.
- Vijay V. Raghavan's Notes/Lecture Material
  - http://www.cacs.louisiana.edu/~cmps561/561/notes/Model.pdf
  - Material in Slides ued with permission