Notes Text Retrieval and Search Engines

Lecture 1
- Deep and shallow NLP
- bag of words

Lecture 2
- pull (search engines / user)
- push (recommender systems)
  - systems take initiative

Lecture 3 Text retrieval Problem
- empirically defined problem
- formula for text retrieval
  - vocabulary, query, document, collection
- query q_i element of vocabulary
- document d_i = d_i1 .. d_ij element of vocabulary
- collection - literally a collection of documents
- set of relevant documents: R(q) subset of collection C
  - query is a “hint” on which document should be in R(q)
- Task = compute R’(q), an approximation of R(q)
- How to compute R’(q)
  - Document selection - function to determine if a document is relevant to the user
  - Document Ranking - function to determine which document is more relevant (relative relevance).
    - what documents are in the approximation set is determined by the threshold

Lecture 4 - Overview of text retrieval methods
- How to design a ranking function
  - query, document, ranking function
  - function must measure the likelihood that document d is relevant to query q
  - retrieval model = formalization of relevance (computational model)
- similarity based models
  - vector space model
- probabilistic models - function = p(R=1|d,q) indicate whether a document is relevant to a query
  - classic probabilistic
  - language model
  - divergence from randomness model
- probabilistic inference model: function = p(d->q) the query follows from the document
- axiomatic model: function must satisfy a set of constraints

- f(q,d)
  - scores depend on the scores of each individual word in the query
  - Use many heuristics
  - How many times does the word appear in the document
  - How long is the document
  - How often does the word appear in the entire collection
    - document frequency

- use a probability of the word in the collection P("door"|C)

Lecture 5 - Vector Space Model

- if a document is more simple to a model than another document then it is assumed that the document has a higher relevance
- represent a doc/query by a term vector
  - term - basic concept (word or a phrase)
  - each term defines one dimension
  - N terms define an N-dimensional space
  - query vector: q = (x_1,x_2) is query term weight
  - doc vector: d = (y_1,y_2) is doc term weight
- relevance(q,d) scale to similarity(q,d)
- How to define/select the basic concept
  - concepts are assumed to be orthogonal
- How to place documents and query in the space (how to assign term weights)
  - term weight in query indicates importance of term
  - term weight in doc indicates how well the term characterizes the doc
- How to define the similarity measure?

Lecture 6 Vector space model

- How to define the dimension?
- How to place a query vector?
- Match similarity ?
- Dimension instantiation: Bag of Words
  - N words in vocabulary therefore there are N dimensions
- Vector placement: Bit vector
  - x_i,y_i element of {0,1} query=(x), document=(y)
- Similarity Instantiation: Dot Product
  - the dot product of two vectors Sim(q,d) = q dot d = x_1 * y_1 … summation
- Simplest VSM = Bit-Vector + Dot-Product + BOW
-