Information Retrieval

(Slides occasionally based on those of Prof. Rao Kambhampati)

Data retrieval with files of text (multimedia)

Functional View

\[
\begin{align*}
&L_{\text{declare/constrain}} \\
&\text{DEQUEUE / CONSTRAIN} \\
&L_{\text{tell}} \quad \text{TELL} \quad L_{\text{ask}} \\
&\text{ASK} \\
&L_{\text{question}} \quad L_{\text{answer}}
\end{align*}
\]

• \(L_{\text{tell}}\): collection of “documents” (unstructured data)
• \(L_{\text{question}}\): user’s “information needs”
• \(L_{\text{answer}}\): collection of “relevant” documents
• query answering spec.: definition of “relevant”, ...

I. Boolean retrieval

\[
\begin{align*}
L_{\text{tell}} & \quad \text{TELL} \quad L_{\text{question}} \\
\text{Info Manager} & \quad \text{ASK} \\
L_{\text{question}} & \quad L_{\text{answer}}
\end{align*}
\]

• \(L_{\text{question}}\): Boolean expression of words
  - e.g., “tiramisu and liqueur and not cake”
• \(L_{\text{answer}}\): collection of “relevant” documents
• Specification of relevant:
  - reduce formula to Disjunctive Normal Form
  \((w_{11} \lor w_{12} \lor \ldots) \lor (w_{21} \lor w_{22} \lor \ldots) \lor (w_{n1} \lor w_{n2} \lor \ldots)\)
  - treat docs as sets of words; return all docs with every word in some conjunct \((w_{ki} \land w_{kj} \land \ldots)\)

Information Retrieval: “normalization”

To address problems 1:

a) Lexical analysis: normalize “words”
  - eliminate hyphens (but MS-DOS?)
  - punctuation marks (but John’s vs Johns, ’03)
  - normalize case of letters (but us vs US)
  - Another problem: users can’t tell what systems has done
  (check out google, altavista, other web search engines and see what they do)

b) Stemming

Identify morphological variants, creating groups
  - system / systems
  - forget / forgetting / forgetful
  - analyse / analysed / analysing / analysis / analytical
Possible uses:
  - replace word by group representative (in document)
  - replace word by all variants in its group (in query)

Well known algorithm by Porter, makes 5 passes; based on condition-action rules (available in public Bow collection you will be using)

IT IS HEURISTIC!!! (because it does not use a dictionary, to make it fast)

Too aggressive
  - organization / organ
  - policy / police
  - army / arm
  - executive / execute

Too timid
  - european / europe
  - cylindrical / cylinder
c) Enriching/normalizing
- Forming compound nouns: ‘computer science’
- Thesaurus:
  - create non-morphologically related group of words (tree):
    - synonyms (arbor),
    - hypernyms - more general/broader than (plant),
    - hyponyms - more specific/narrower than (sapling)
  e.g., 
  § Roget’s Thesaurus - more useful for literature
  § WordNet [Miller] - become widely used as a simple ontology
- Domain-specific thesaurus:
  - more powerful: terminology of comp science
  - automatically generated thesaurus from the given document corpus: based on correlated occurrence of terms in the same “context” (what is context: document, paragraph, sentence structure)?; can work statistically when there are MANY documents

II. Boolean Retrieval with Controlled Vocab.
- Alternative approach to “information needs” problem:
  describe what text is about
e.g., rather than view text as a collection of its words, assign to each document a small collection of words from a controlled vocabulary (e.g., the NASA thesaurus for the aerospace discipline, the MESH thesaurus for medicine, CACM/dmoz/yahoo subject hierarchy,) representing its content
  $L_{tell} = \langle \text{doc}, \langle \text{keyword1,keyword2,...} \rangle \rangle$
- Who annotates the documents?
  - author, librarian, machine (semi) automatic classifier, possibly based on machine learning

Approximate Query Answering
- Note that all the previous steps are heuristic: they may improve answers, but occasionally they can cause problems (e.g., introduce additional ambiguity)
- So we are giving up on the idea of “perfect data answer”, as in databases, in order to get better “information answer”
- Additional possibilities:
  - provide ranked list of answers: present first those most sure to be of interest to the user; this addresses problem #3 (“too many answers”)
  - co-operative answering (e.g., iterative refinement, automatic weakening when answer set is empty)

Alternate Model of IR
- $L_{tell} \xrightarrow{TELL} \text{Info Manager}$
- $L_{question} \xrightarrow{ASK} \text{Lanswer}$
- $L_{question}$: another (unstructured) document, even if it is short (e.g., English sentence): describes user interests
- $L_{answer}$: ranked/ordered list of relevant documents
- Specification of “ranking” (and hence “relevance”): based on a similarity function $\text{sim}(\text{doc}, \text{query})$

II. Vector Space model of similarity
- Document = set of words/index terms.
  - represent collection as term/document boolean matrix $W_{[k,j]}$
- What are reasonable models of “similarity” in this case?
  Think of each document as vector in n-dimensional space of index terms. Mathematicians have studied lots of “distance measures”
  - “Euclidean distance”:
    $\sqrt{\Sigma (w_{k}[j] - q[j])^2}$
  - “Dot product” $W_{k} \cdot Q = \Sigma (w_{k}[j] * q[j])$
- Intuitively, want measures that
  - allow partial match
  - favor documents with more words in common
  - have bounded value (e.g., between 0 and 1)
  - favor documents in which shared words occur more often! (=> treat document as bag of words, and matrix as count of words)
Vector Space model of similarity - 2

- Document as bag of terms: term frequency matrix $W[k,j]$
  - a: System and human system engineering testing of EPS
  - b: A survey of user opinion of computer system response time
  - c: The EPS user interface management system
  - d: Human machine interface for ABC computer applications
  - e: Relation of user perceived response time to error measurement
  - f: The generation of random, binary, ordered trees
  - g: The intersection graph of paths in trees
  - h: Graph minors IV: Widths of trees and well-quasi-ordering

  Interface 001000001
  User 011010001
  System 111000001
  Human 100100000
  Computer 010200000
  Response 010010000
  Time 010010000
  EPS 101000000
  Survey 010000000
  Trees 000001110
  Graph 000000110
  Minors 000000010

- Hence document vector is no longer binary

- TF-IDF
  - The following formula has been developed empirically
  \[
  \text{TF-IDF}_j = (0.5 + \frac{0.5 \times \text{freq}(j,k)}{\max\{\text{freq}(j,k)\}}) \times \log(N/n_j)
  \]
  - For the query document $q$, it is suggested to use $w_q[j]$ to rank answers

- Evaluation of TF-IDF
  - Seems to be the “golden standard”
  - nothing else proposed has been uniformly better!!!
  - In general, vector-space models assume that index terms are independent "orthonormal" so they can be used as axes for vector space
  - this is unreasonable, but seems to work!!

- III. Probabilistic model of IR
  - Vector space model is "unprincipled"
  - Probabilistic model principle: given set of docs $\{d_j\}$ & query $q$, try to estimate the probability $\Pr(\text{Relevant} \mid q, d_j)$, and return documents ranked by this probability. [Some use notation $Pr(\text{Relevant} \mid q, d_j)$, or just $Pr(\text{Relevant} \mid d_j)$, $q$ being understood]
Probabilistic model
Given R, if we studied hard all the docs, we might find features (like feature vectors) that are associated with (not) being a member of R (called ‘classification’) i.e., Pr(wk is relevant | R). So, would like to express Pr(R|x) in terms of Pr(x|R).

Use Bayes’ formula: Pr(R|x) = Pr(x|R) * Pr(R)/Pr(x), and binary independence assumption Pr({t1,t3}|R) = Pr(t1|R) · Pr(t2|R), plus some math to arrive at a general formula for sim, which uses only – Pr(tj | R) -- prob. that feature tj is present in a relevant document – Pr(tj |~R) -- prob. that feature tj is present in an irrelevant document

How to estimate these:
initially: let nj=#(docs containing tj) , N=#docs; approx. by Pr(tj|R) by 0.5, Pr(tj|~R) by nj/N

feedback: - retrieve subset V of r docs; assume all are relevant, and others are irrelevant (!?); - let Tj=(docs in V having feature tj); now approximate Pr(tj|R) by #Tj / #V, Pr(tj|~R) by (nj - #Tj) / (N- #V)

probabilistic bayesian networks
- Links as dependencies: conditional probability table for each node, based on parents (gives “influences”/”causes”) -- see below
- Independence assumptions: variables not connected by link are not directly conditions e.g., Pr(thunder | storm,lightning,campfire) = Pr(thunder | lightning)

probabilistic bayesian networks
- Allows computation of full joint probability Pr(S,B,L,C,T,F) in terms of Pr(node y | parents(y) ): Pr(y1,...,yn) = \prod Pr(yi | parents(yi))
- All that is required is that for each node xi, we have “influence” functions Fi such that 0<Fi(xi,parents(xi))<1 and \sum Fi(xi,parents(x)) = 1

plus “prior” distrib’n for root nodes.

BN-based IR model
- Document Network: Large, but computed once for each document collection
- Query Network: Small, computed for every query

Example
- Document Network
- Query Network

Probabilistic Bayesian Networks
- ...
Simple BN-based IR model

- Ranking of doc_k with respect to query q is measured by evidence(prob.) for q provided by observation of that doc
- Draw BT graph, where terms separate docs from query q
  \[ \text{Pr}(\text{doc}_k \cap q) = \sum \text{Pr}(\text{doc}_k \cap q \mid w_j) \times \text{Pr}(w_j) \] where w_j are binary term vectors
  \[ \text{Pr}(\text{doc}_k \mid q) = \sum \text{Pr}(q \mid \text{doc}_k \cap w_j) \times \text{Pr}(w_j \mid \text{doc}_k) \]

Can replace \( \text{Pr}(w_j \mid \text{doc}_k) \) by \( \sum \text{Pr}(t_i \mid \text{doc}_k) \) for \( t_i \) appearing in \( w_j \)

BN based IR model

Choice of probabilities can “simulate” various models:

- Boolean (conjunctive only) model:
  - priors: all docs equally likely, 1/N
  - \( \text{Pr}(t_j \mid \text{doc}_k) = 1 \) if \( t_j \) appears in \( \text{doc}_k \) (0 otherwise)
  - \( \text{Pr}(q \mid \text{doc}) = 1 \) if \( q \) appears in \( \text{doc}_k \) (0 otherwise)

- tf-ifd model (close to it):
  - priors \( \text{Pr}(\text{doc}) \) reflect normalization = \( 1/|\text{doc}_k| \)
  - \( \text{Pr}(t_j \mid \text{doc}_k) = \text{tf}_k[j] \) if \( t_j \) appears in \( \text{doc}_k \) (0 otherwise)
  - \( \text{Pr}(q \mid c) = \text{idf}[j] \) if \( c = t_j \) and \( t_j \) appears in \( q \) (0 otherwise)

More importantly, this model supports the combination of multiple criteria/techniques for specifying same query

IV. InfoRetr with Relevance Feedback

- L_tell : Previous answer list annotated by human with +, -
- Idea: improve notion of “relevance” being used for that query

Relevance feedback for vector model

- Can be shown that if you knew complete set of relevant documents, the optimal query for it would be
  \[ Q_{\text{opt}} = \sum \text{Pr}(q \cap \text{doc}_k) \mid q \cap \text{doc}_k \]

Rocchio method
  \[ Q_i = \alpha q_0 + \beta \sum_j \text{Pr}(t_j \mid q_0) - \gamma \sum_j \text{Pr}(t_j \mid \text{doc}_k) \]
  \( q_0 \) is initial query. \( Q_i \) is “improved query”

- \( D_R \) is set of docs retrieved marked relevant by user
- \( D_{\text{irrelevant}} \) is set of irrelevant docs retrieved
  \( \alpha = 1; \beta = .75; \gamma = .25 \) typically.

- So, terms in original query are “reweighted”, and query is “expanded” with terms appearing in relevant documents, and somewhat “trimmed” of terms in irrelevant documents

- Simple, gives reasonable results empirically, but unprincipled

IV. InfoRetr with Relevance Feedback

Measuring Performance

- Precision
  \[ \text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \]
  - Proportion of selected items that are correct

- Recall
  \[ \text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \]
  - Proportion of target items that were selected

- Precision-Recall curve
  - But a system could return just 1 doc, sure to be right!?
  - Precision vs Recall curve
Precision/Recall Curves

11-point recall-precision curve

Example: Suppose for a given query, 10 documents are relevant (in blue below). Suppose when all documents are ranked in descending similarities, we have

d_1 \ d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7 \ d_8 \ d_9 \ d_{10} \ d_{11} \ d_{12} \ d_{13} \ d_{14} \ d_{15} \ d_{16} \ d_{17} \ d_{18} \ d_{19} \ d_{20} \ d_{21} \ d_{22} \ d_{23} \ d_{24} \ d_{25} \ d_{26} \ d_{27} \ d_{28} \ d_{29} \ d_{30} \ d_{31} \ ...

Precision Recall Curves…

When evaluating the retrieval effectiveness of a text retrieval system or method, a large number of queries are used and their average 11-point recall-precision curve is plotted.

- Methods 1 and 2 are better than method 3.
- Method 1 is better than method 2 for high recalls.