Artificial Intelligence -- 15-381
Information Retrieval Methods

Jaime Carbonell
7-March-2000

OUTLINE:

- The IR task (query, doc collections,...)
- The vector space formalism (VSM)
- Term weighting methods (TfIDf)
- Query expansion and relevance feedback
- Beyond VSM: GVSM, MMR, ...

Information Retrieval: The Challenge

Text DB includes:

(1) Rainfall measurements in the Sahara continue to show a steady decline starting from the first measurements in 1961. In 1996 only 12mm of rain were recorded in upper Sudan, and 1mm in Southern Algiers...

(2) Dan Marino states that professional football risks loosing the number one position in heart of fans across this land. Declines in TV audience ratings are cited...

(3) Alarming reductions in precipitation in desert regions are blamed for desert encroachment of previously fertile farmland in Northern Africa. Scientists measured both yearly precipitation and groundwater levels...

User query states:

"Decline in rainfall and impact on farms near Sahara"

Challenges

- How to retrieve (1) and (3) and ***not*** (2)?
- How to rank (3) as best?
- How to cope with no shared words?

Information Retrieval Assumptions

Basic IR task

- There exists a document collection $\{D_j\}$
- Users enters *at hoc* query Q
- Q correctly states user's interest
- User wants subset $\{D_i\} < \{D_j\}$ most relevant to Q
- Or, user wants relevance ranked prefix of $\{D_j\}$

"Shared Bag of Words" assumption

Every query = $\{w_i\}$
Every document = $\{w_k\}$
...where w_i & w_k in same Σ

All syntax is irrelevant (e.g. word order)
All document structure is irrelevant
All meta-information is irrelevant
(e.g. author, source, genre)
=> Words suffice for relevance assessment

Retrieval by shared words

If Q and D_j share ***some*** w_i,
then Relevant(Q,D_j)

If Q and D_j share ***all*** w_i, then Relevant(Q,D_j)

If Q and D*j* share over K% of w_i,
then Relevant(Q,D_j)

Boolean Queries

Industrial use of Silver

Q: *silver*

```
R: "The Count’s silver anniversary..."
   "Even the crash of ’87 had a silver lining..."
   "The Lone Ranger lived on in syndication..."
   "Sliver dropped to a new low in London..."
   ...
```

Q: *silver AND photography*

```
R: "Posters of Tonto and the Lone Ranger..."
   "The Queen’s Silver Anniversary photos..."
   ...
```

Q: *(silver AND (NOT anniversary)*
AND (NOT lining)
AND emulsion)
OR (AgI AND crystal
AND photography))

```
R: "Silver Iodide Crystals in Photography..."
   "The emulsion was worth its weight in silver..."
   ...
```

Boolean queries are:

a) easy to implement
b) confusing to compose
c) seldom used (except by librarians)
d) prone to low recall
e) all of the above

Beyond the Boolean Boondoggle

Desiderata

- Query must be ***natural*** for all users
  - Sentence, phrase, or word(s)
  - No AND’s, OR’s, NOT’s, ...
  - No parentheses (no structure)
- System focus on important words
  - Q: *I want* **laser printers** *now*
- Find what I mean, not just what I say Q: *cheap car insurance*

```
(pAND (pOR
        "cheap" [1.0]
         "inexpensive" [0.9]
         "discount" [0.5)]
      (pOR "car" [1.0]
           "auto" [0.8]
           "automobile" [0.9]
           "vehicle" [0.5])
      (pOR "insurance" [1.0]
           "policy" [0.3]))
```

- Speech-recognized queries
  - Coming soon, to a system near you
  - longer queries
  - more fluff words
  - acoustic recognition errors

The Vector Space Model

Let Σ = [w_1, w_2, ... w_n]

Let D_j = [$c(w_1,D_j)$, $c(w_2,D_j)$, ... $c(w_n,D_j)$]

Let Q = [$c(w_1,Q)$, $c(w_2,Q)$, ... $c(w_n,Q)$]

Initial Definition of Similarity:

$S_I(Q,D_j) = Q \cdot D_j$

Normalized Definition of Similarity:

$S_N(Q,D_j) = (Q \cdot D_j)/(|Q| \times |D_j|)$
$= \cos(Q,D_j)$

Relevance Ranking

If $S(NQ,D_i) > S(NQ,D_j)$
Then D_i is more relevant than D_i to Q

Retrieve(k,Q,{D_j} =
$Arg\max^k[\cos(Q,D_j)]$
D_j in {D_j}

A Refinement to VSM

Word normalization

- Words in morphological root form
  countries => country
  interesting => interest
- Stemming as a fast approximation
  countries, country => countr
  interesting => interest moped => mop
- Reduces vocabulary (always good)
- Generalizes matching (usually good)
- More useful for non-English IR
  (Arabic has > 100 variants per verb)

More Refinements to VSM

Stop-Word Elimination

- Discard articles, auxiliaries, prepositions, ... typically 100-300 most frequent small words
- Reduce document length by 30-40%
- Retrieval accuracy improves slightly (5-10%)

Proximity Phrases

- E.g.: "air force" => airforce
- Found by high-mutual information
  $p(w_1 w_2) \gg p(w_1)p(w_2)$
  $p(w_1 \& w_2$ in k-window$) \gg p(w_1$ in k-window$)p(w_2$ in same k-window$)$
- Retrieval accuracy improves slightly (5-10%)
- Too many phrases => inefficiency

Words => Terms

- term = word | stemmed word | phrase
- Use exactly the same VSM method on terms (vs words)

Evaluating Information Retrieval

$$\text{Recall} = \frac{\text{retrieved \& relevant}}{\text{all relevant docs}}$$

$$\text{Precision} = \frac{\text{retrieved \& relevant}}{\text{all retrieved docs}}$$

Contingency table:

	relevant	not-relevant
retrieved	a	b
not retrieved	c	d

P = a/(a+b)   R = a/(a+c)

Accuracy = (a+d)/(a+b+c+d)

F1 = 2PR/(P+R)

Miss = c/(a+c) = 1 - R   (false negative)

F/A = b/(a+b+c+d)   (false positive)

11-point precision curves

- IR system generates total ranking
- Plot precision at 10%, 20%, 30% ... recall,

Query Expansion

Observations:

- Longer queries often yield better results
- User's vocabulary may differ from document vocabulary

  `Q:` *how to avoid heart disease*

  ```
  D: "Factors in minimizing stroke and
      cardiac arrest: Recommended dietary
      and exercise regimens"
  ```

- Maybe longer queries have more chances to help recall.

Bridging the Gap

- Human query expansion (user or expert)
- Thesaurus-based expansion
  Seldom works in practice (unfocused)
- Relevance feedback
  - Widen a thin bridge over vocabulary gap
  - Adds words from document space to query
- Pseudo-Relevance feedback
- Local Context analysis

Relevance Feedback

Rocchio Formula

$Q' = Q + D_{ret}$

$W(t,Q') = \alpha W(t,Q) + \beta W(t,D_{rel}) - \gamma W(t,D_{irr})$

Relevance Feedback Process

1. Ranked-list retrieval: $R(Q,\{D_j\})$
2. User selects D_{rel} and D_{irr}
   from among the top k retrieved documents
3. Q'_{RF} recalculated using Rocchio (above)
4. Re-retrieval: $R(Q'_{RF},\{D_j\})$
5. Optional: 2nd-pass Rocchio (go to step 2)

Pseudo-Relevance Feedback

1. Ranked-list retrieval: $R(Q,\{D_j\})$
2. System assumes top-k are relevant (very small k)
3. Q'_{PRF} recalculated using α and β terms only
4. Re-retrieval: $R(Q'_{PRF},\{D_j\})$
5. Never second pass (why not?)

Term Weighing in IR

Definitions

w_i:	"ith Term:" a word, stemmed word, or indexed phrase
D_j:	"jth Document:" a unit of indexed text, e.g. a web-page, a news report, an article, a patent, a legal case, a book, a chapter of a book, etc.
C:	"The Collection:" the full set of indexed documents
$Tf(w_i,D_j$:	"Term Frequency:" the number of times w_i occurs in document D_j. Tf is sometimes normalized by dividing by frequency of the most-frequent non-stop term in the document [$Tf_{norm} = Tf/Tf_{max}$].
$Df(w_i,C)$:	"Document Frequency:" the number of documents from C in which w_i occurs. Df is usually normalized by dividing it by the total number of documents in C, i.e. the size(C).
$IDf(w_i,C)$:	"Inverse Document Frequency:" $[Df(w_i,C)/size(C)]^{-1}$. Most often the $log_2(IDf)$ is used, rather than IDf directly.

TfIDf Term Weights

In general: $TfIDf(w_i,D_j,C) =$
$F_1(Tf(w_i,D_j) * F_2(IDf(w_i,C))$

Usually $F_1 = 0.5 + log_2(Tf)$, or Tf/Tf_{max}

or $0.5 + 0.5Tf/Tf_{max}$

Usually $F_2 = \log_2(IDf)$

In the SMART system: $TfIDf(w_i,D_j,C) =$
$[0.5 + 0.5Tf(w_i,D_j/Tf_{max}(D_j)] * \log_2(IDf(w_i,C))$

Term Weighting beyond TfIDf

Probabilistic Models

- Old style (see textbook)
  Improves precision-recall slightly
- IBM/CMU Channel model (see Berger lecture)
  Improves precision-recall significantly

Neural Networks

- Theoretically attractive
- Do not scale up, unfortunately

Fuzzy Sets

- Not deeply researched (see textbook)

Natural Language Analysis

- Analyze and understand D's & Q first
- Ultimate IR method, in theory
- Generally NL analysis is an unsolved problem
- Scale up challenges, even if we could it
- But, improves IR for very limited domains

Generalized Vector Space Model

Principles

- Define terms by their occurrence patterns in documents
- Define query terms in the same way
- Compute similarity by document-pattern overlap for terms in D and Q
- Use standard Cos similarity and either binary or TfIDf weights

Advantages

- Automatically calculates partial similarity
  If "heart disease" and "stroke" and "ventricular" co-occur in many documents, then if the query contains only one of these terms, documents containing the other will receive partial credit proportional to their document co-occurrence ratio.
- No need to do query expansion or relevance feedback

Disadvantages

- Computationally expensive
- Performance = vector space + Q expansion

A Critique of Pure Relevance

Non-Redundancy in IR

- Ideal: Find ***Relevant*** and ***Novel*** info
- State of practice: ***Relevant*** only
- State of the research: ***Relevant*** and ***Not Redundant***

Operationalize "Anti-Redundancy"

- Retrieve "new" or "different" documents
- System ***must know*** what it retrieved before
- System ***might know*** what the user already knows
- Maximize similarity(D_i,Q) as before, but...
  Minimize similarity(D_i, D_{known})
  In a single utility metric to rank D_i's
- Metric must be dynamic, responding to user's increase in knowledge as she reads retrieved doc's.

Relevance => Marginal Relevance

- How much new relevant info does this doc add?
- Want to Maximize Marginal Relevance (MMR)

Open Research Problems in IR

Beyond VSM

- Vectors in different Spaces:
  Generalized VSM, Latent Semantic Indexing...
- Probabilistic IR (Language Modeling):
  $P(D|Q) = P(Q|D)P(D)/P(Q)$

Beyond Relevance

- Appropriateness of doc to user
  comprehension level, etc.
- Novelty of information in doc to user
  anti-redundancy as approx to novelty

Beyond one Language

- Translingual IR
- Transmedia IR

Beyond Content Queries

- "What's new today?"
- "What sort of things to you know about"
- "Build me a Yahoo-style index for X"
- "Track the event in ***this*** news-story"