CS-490W
Web Information Retrieval & Management

Basic Concepts of Information Retrieval

Luo Si

Department of Computer Science
Purdue University
Basic Concepts of IR: Outline

Basic Concepts of Information Retrieval:

- Task definition of Ad-hoc IR
  - Terminologies and concepts
  - Overview of retrieval models

- Text representation
  - Indexing
  - Text preprocessing

- Evaluation
  - Evaluation methodology
  - Evaluation metrics
Ad-hoc IR: Terminologies

Terminologies:

- **Query**
  - Representative data of user’s information need: text (default) and other media

- **Document**
  - Data candidate to satisfy user’s information need: text (default) and other media

- **Database|Collection|Corpus**
  - A set of documents

- **Corpora**
  - A set of databases
  - Valuable corpora from TREC (Text Retrieval Evaluation Conference)
Ad-hoc IR: Introduction

Ad-hoc Information Retrieval:

- Search a collection of documents to find relevant documents that satisfy different information needs (i.e. queries)
- Example: Web search
Ad-hoc IR: Introduction

Ad-hoc Information Retrieval:
- Search a collection of documents to find relevant documents that satisfy different information needs (i.e., queries)

- Queries are created and used dynamically; change fast
- “Ad-hoc”: formed or used for specific or immediate problems or needs” – Merriam-Webster’s collegiate Dictionary

Ad-hoc IR vs. Filtering
- Filtering: Queries are stable (e.g., Asian High-Tech) while the collection changes (e.g., news)
- More for filtering in later lectures
Content Based Filtering Filtering

Information Needs are Stable
System should make a delivery decision on the fly when a document “arrives”

User Profile: Asian High-Tech
AD-hoc IR: Basic Process

1. Information Need
2. Representation
3. Query
4. Retrieval Model
5. Retrieved Objects
6. Evaluation/Feedback
7. Representation
8. Indexed Objects
AD-hoc IR: Overview of Retrieval Model

Retrieval Models

- Boolean
- Vector space
  - Basic vector space
  - Extended Boolean
- Probabilistic models
  - Statistical language models
  - Two Possion model
  - Bayesian inference networks
- Citation/Link analysis models
  - Page rank
  - Hub & authorities

SMART, LUCENE
Lemur
Okapi
Inquery
Google
Clever
AD-hoc IR: Overview of Retrieval Model

Retrieval Model

Determine whether a document is relevant to query

- Relevance is difficult to define
  - Varies by judgers
  - Varies by context (i.e., jointly by a set of documents and queries)

- Different retrieval methods estimate relevance differently
  - Word occurrence of document and query
  - In probabilistic framework, $P(\text{query}|\text{document})$ or $P(\text{Relevance}|\text{query,document})$
  - Estimate semantic consistency between query and document
Types of Retrieval Models

- Exact Match (Document Selection)
  - Example: Boolean Retrieval Method
  - Query defines the exact retrieval criterion
  - Relevance is a binary variable; a document is either relevant (i.e., match query) or irrelevant (i.e., mismatch)
  - Result is a set of documents
    - Documents are unordered
    - Often in reverse-chronological order (e.g., Pubmed)
Types of Retrieval Models

- Best Match (Document Ranking)
  - Example: Most probabilistic models
  - Query describes the desired retrieval criterion
  - Degree of relevance is a continuous/integral variable; each document matches query to some degree
  - Result in a ranked list (top ones match better)
    - Often return a partial list (e.g., rank threshold)

```
Best Match

Doc1 0.99  +
Doc2 0.90  +
Doc3 0.85  +
Doc4 0.82  -
Doc5 0.81  +
Doc6 0.79  -
```

Rank
Types of Retrieval Models

Exact Match (Selection) vs. Best Match (Ranking)

- Best Match is usually more accurate/effective
  - Do not need precise query; representative query generates good results
  - Users have control to explore the rank list: view more if need every piece; view less if need one or two most relevant

- Exact Match
  - Hard to define the precise query; too strict (terms are too specific) or too coarse (terms are too general)
  - Users have no control over the returned results
  - Still prevalent in some markets (e.g., legal retrieval)
AD-hoc IR: Overview of Retrieval Model

Retrieval Models

- Boolean

- Vector space
  - Basic vector space
  - Extended Boolean

- Probabilistic models
  - Statistical language models
  - Two Possion model
  - Bayesian inference networks

- Citation/Link analysis models
  - Page rank
  - Hub & authorities
AD-hoc IR: Basic Process

- Information Need
- Representation
  - Query
  - Retrieval Model
  - Retrieved Objects
  - Evaluation/Feedback
- Indexed Objects
It never leaves my side, April 6, 2002

Reviewer: "dage456" (Carmichael, CA USA) - See all my reviews

It fits in the palm of your hand and is the size of a deflated wallet (wonder where the money went). I have had my ipod now for 4 months and cannot imagine how I used to get by with my old rio 600 with is 64 megs of ram and.. usb connection. Because of its size this little machine goes with my everywhere and its ten hour battery life means I can listen to stuff all day long.

Pros: size, both physical and capacity.
design: It looks beautiful
controls: simple and very easy to use
connection: FIREWIRE!!

Cons: needs the ability to bookmark. I use my ipod mostly for audiobooks. the ipod needs to include a bookmark feature for those like me.

From Amazon Customer Review of IPod
What computer see

See all my reviews</a></td></tr></table>

It fits in the palm of your hand and is the size of a deflated wallet (wonder where the money went). I have had my ipod now for 4 months and cannot imagine how I used to get by with my old rio 600 with is 64 megs of ram and.. usb connection. Because of its size this little machine goes with my everywhere and its ten hour battery life means I can listen to stuff all day long.

Pros: size, both physical and capacity.<br>design: It looks beautiful<br>controls: simple and very easy to use<br>connection: FIREWIRE!!

Cons: needs the ability to bookmark. I use my ipod mostly for audiobooks. the ipod needs to include a bookmark feature for those like me.

From Amazon Customer Review of IPod
Iran's government is intensifying a birth control program _ despite opposition from radicals _ because the country's fast-growing population is imposing strains on a struggling economy.

.............

</TEXT>
</DOC>
Indexing

Associate document/query with a set of keys

- Manual or human Indexing
  - Indexers assign keywords or key concepts (e.g., libraries, Medline, Yahoo!); often small vocabulary
  - Significant human efforts, may not be thorough

- Automatic Indexing
  - Index program assigns words, phrases or other features; often large vocabulary
  - No human efforts
Controlled Vocabulary vs. Full Text

- Controlled Vocabulary Indexing
  - Assign words from a small vocabulary or a node from an ontology
  - Often manually but can be done by learning algorithms

- Full Indexing:
  - Often index with an uncontrolled vocabulary of full text
  - Automatically while good algorithm can generate more representative keywords/ key concepts
Mutation of a mutL homolog in hereditary colon cancer.


Johns Hopkins Oncology Center, Baltimore, MD 21231.

Some cases of hereditary nonpolyposis colorectal cancer (HNPCC) are due to alterations in a mutS-related mismatch repair gene. A search of a large database of expressed sequence tags derived from random complementary DNA clones revealed three additional human mismatch repair genes, all related to the bacterial mutL gene. One of these genes (hMLH1) resides on chromosome 3p21, within 1 centimorgan of markers previously linked to cancer susceptibility in HNPCC kindreds. Mutations of hMLH1 that would disrupt the gene product were identified in such kindreds, demonstrating that this gene is responsible for the disease. These results suggest that defects in any of several mismatch repair genes can cause HNPCC.
MeSH Tree Structures

1. Anatomy [A]
2. Organisms [B]
3. Diseases [C]
   - Bacterial Infections and Mycoses [C01]
   - Virus Diseases [C02]
   - Parasitic Diseases [C03]
   - Neoplasms [C04]
   - Musculoskeletal Diseases [C05]
   - Digestive System Diseases [C06]

   Neoplasms by Site
   - Digestive System Neoplasms
     - Gastrointestinal Neoplasms
       - Intestinal Neoplasms
         - Colorectal Neoplasms
           - Colorectal Neoplasms, Hereditary Nonpolyposis

4. Chemicals and Drugs [D]
5. Analytical, Diagnostic and Therapeutic Techniques
6. Psychiatry and Psychology [F]
7. Biological Sciences [G]
8. Physical Sciences [H]
PMID- 8128251
TI - Mutation of a mutL homolog in hereditary colon cancer.
MH - *Adenosinetriphosphatase
MH - Amino Acid Sequence
MH - Bacterial Proteins/chemistry/*genetics
MH - Base Sequence
MH - Carrier Proteins
MH - Chromosome Mapping
MH - *Chromosomes, Human, Pair 3
MH - Codon
MH - Colorectal Neoplasms, Hereditary Nonpolyposis/*genetics
MH - *DNA Repair
MH - *DNA-Binding Proteins
Pros and cons of controlled vocabulary indexing

- **Advantages**
  - Many available vocabularies/ontologies (e.g., MeSH, Open Directory, UMLS)
  - Normalization of indexing terms: less vocabulary mismatch, more consistent semantics
  - Easy to use by RDBMS (e.g., semantic Web)
  - Support concept based retrieval and browsing

- **Disadvantages**
  - Substantial efforts to be assigned manually
  - Inconvenient for users not familiar with the controlled vocabulary
  - Coarse representation of semantic meaning
Text Representation: Indexing

Full Text Indexing

Full text Indexing: index all text with uncontrolled vocabulary

- Advantages
  - (Possibly) Keep all the information within the text
  - Often no human efforts; easy to build

- Disadvantages
  - Difficult to cross vocabulary gap (e.g., “cancer” in query, “neoplasm” in document)
  - Large storage space

How to build full text Indexing:
  - What are the candidates in the word vocabulary? Are they effective to represent semantic meanings
  - How to bridge small vocabulary gap (e.g., car and cars)
### Text Representation: Indexing

#### Statistical Properties of Text

<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency</th>
<th>Word</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>1130021</td>
<td>market</td>
<td>52110</td>
</tr>
<tr>
<td>of</td>
<td>547311</td>
<td>bank</td>
<td>47940</td>
</tr>
<tr>
<td>to</td>
<td>516636</td>
<td>stock</td>
<td>47401</td>
</tr>
<tr>
<td>a</td>
<td>464736</td>
<td>trade</td>
<td>47310</td>
</tr>
<tr>
<td>in</td>
<td>390819</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>and</td>
<td>387703</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>....</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Statistics collected from Wall Street Journal (WSJ), 1987
Text Representation: Indexing
Statistical Properties of Text

Term
Frequency

Term Rank

Term Frequency
Observations from language/corpus independent features

- A few words occur very frequently (High Peak)
  - Top 2 words: 8%-15% (e.g., words that carry no semantic meanings like “the”, “to”)

- Most words occur rarely (Heavy Tail)

- Representative words often in the middle
  - e.g., market and stock for WSJ

- Rules formally describe word occurrence patterns:
  Zipf’s law, Heaps’ Law
Zipf’s law: relate a term’s frequency to its rank

- Rank all terms with their frequencies in descending order, for a term at a specific rank (e.g., r) collects and calculates

\[ p_r = \frac{f_r}{N} \]

: relative term frequency

\[ f_r : \text{term frequency} \]

- Zipf’s law (by observation):

\[ p_r = \frac{A}{r} \quad A \approx 0.1 \]

So \[ p_r = \frac{f_r}{N} = \frac{A}{r} \Rightarrow rf_r = AN \Rightarrow \log(r) = -\log(f_r) + \log(AN) \]

So Rank X Frequency = Constant
Text Representation: Indexing
Statistical Properties of Text

Term Frequency

Term Rank
### Text Representation: Indexing

**Statistical Properties of Text**

<table>
<thead>
<tr>
<th>Word</th>
<th>Frequency</th>
<th>$r \times p_r$</th>
<th>Word</th>
<th>Frequency</th>
<th>$r \times p_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>1130021</td>
<td>0.059</td>
<td>market</td>
<td>52110</td>
<td>0.101</td>
</tr>
<tr>
<td>of</td>
<td>547311</td>
<td>0.058</td>
<td>bank</td>
<td>47940</td>
<td>0.109</td>
</tr>
<tr>
<td>to</td>
<td>516636</td>
<td>0.082</td>
<td>stock</td>
<td>47401</td>
<td>0.110</td>
</tr>
<tr>
<td>a</td>
<td>464736</td>
<td>0.098</td>
<td>trade</td>
<td>47310</td>
<td>0.112</td>
</tr>
<tr>
<td>in</td>
<td>390819</td>
<td>0.103</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>and</td>
<td>387703</td>
<td>0.122</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Statistics collected from Wall Street Journal (WSJ), 1987
Text Representation: Text Preprocessing

Text Preprocessing: extract representative index terms

- Parse query/document for useful structure
  - E.g., title, anchor text, link, tag in xml.....
- Tokenization
  - For most western languages, words separated by spaces; deal with punctuation, capitalization, hyphenation
  - For Chinese, Japanese: more complex word segmentation…
- Remove stopwords: (remove “the”, “is”,..., existing standard list)
- Morphological analysis (e.g., stemming):
  - Stemming: determine stem form of given inflected forms
- Other: extract phrases; decompounding for some European languages “rörelseuppskattningssökningsintervallsinställningar”
## Text Representation: Text Preprocessing

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
<th>Word</th>
<th>Count</th>
<th>Word</th>
<th>Count</th>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>4</td>
<td>at</td>
<td>1</td>
<td>different</td>
<td>1</td>
<td>may</td>
<td>1</td>
</tr>
<tr>
<td>and</td>
<td>3</td>
<td>basal</td>
<td>1</td>
<td>exchange</td>
<td>1</td>
<td>nontarget</td>
<td>1</td>
</tr>
<tr>
<td>by</td>
<td>3</td>
<td>be</td>
<td>1</td>
<td>exogenous</td>
<td>1</td>
<td>not</td>
<td>1</td>
</tr>
<tr>
<td>steroids</td>
<td>3</td>
<td>been</td>
<td>1</td>
<td>fluorescent</td>
<td>1</td>
<td>may</td>
<td>1</td>
</tr>
<tr>
<td>centrioles</td>
<td>2</td>
<td>bodies</td>
<td>1</td>
<td>from</td>
<td>1</td>
<td>of</td>
<td>1</td>
</tr>
<tr>
<td>in</td>
<td>2</td>
<td>can</td>
<td>1</td>
<td>growth</td>
<td>1</td>
<td>precise</td>
<td>1</td>
</tr>
<tr>
<td>affect</td>
<td>1</td>
<td>at</td>
<td>1</td>
<td>has</td>
<td>1</td>
<td>receptor</td>
<td>1</td>
</tr>
<tr>
<td>already</td>
<td>1</td>
<td>cell</td>
<td>1</td>
<td>identity</td>
<td>1</td>
<td>regularly</td>
<td>1</td>
</tr>
<tr>
<td>Although</td>
<td>1</td>
<td>cells</td>
<td>1</td>
<td>level</td>
<td>1</td>
<td>reveal</td>
<td>1</td>
</tr>
<tr>
<td>antibodies</td>
<td>1</td>
<td>cilia-bearing</td>
<td>1</td>
<td>localization</td>
<td>1</td>
<td>Specific</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

24 stopwords out of total 61 words
Text Representation: Bag of Words

The simplest text representation: “bag of words”

- Query/document: a bag that contains words in it
- Order among words is ignored

<table>
<thead>
<tr>
<th>steroids</th>
<th>substance</th>
<th>growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>centrioles</td>
<td>..........</td>
<td>...</td>
</tr>
<tr>
<td>..........</td>
<td>exchange</td>
<td>steroids</td>
</tr>
<tr>
<td>..........</td>
<td>nontarget</td>
<td></td>
</tr>
<tr>
<td>step</td>
<td>bodies</td>
<td>two</td>
</tr>
<tr>
<td>precise</td>
<td>..........</td>
<td></td>
</tr>
</tbody>
</table>

| 3 steroids | 1 cilia-bearing | 1 precise | 1 two |
| 2 centrioles | 1 different | 1 receptor | 1 unexpected |
| 1 affect | 1 exchange | 1 regularly | 1 vitally |
| 1 already | 1 exogenous | 1 reveal | 1 way |
| 1 Although | 1 fluorescent | 1 Specific |
| 1 antibodies | 1 growth | 1 step |
| 1 basal | 1 identity | 1 substance |
| 1 bodies | 1 level | 1 suggests |
| 1 cell | 1 localization | 1 target |
| 1 cells | 1 nontarget | 1 technique |
Text Representation: Phrases

- Single word/stem indexing may not be sufficient
e.g., “hit a home run yesterday”

- More complicated indexing includes phrases (thesaurus classes)

- How to automatically identify phrases
  - Dictionary
  - Find the most common N word phrases by corpus statistics (be careful of stopwords)
  - Syntactic analysis, noun phrases
  - More sophisticated segmentation algorithm like “Hidden Markov Model”
Word Stemming

- Associate morphological variants of words into a single form
  - E.g., plurals, adverbs, inflected word forms
  - May lose the precise meaning of a word

- Different types of stemming algorithms
  - Rule-based systems: Porter Stemmer, Krovetz Stemmer
    - Porter Stemmer Example: describe/describes -> describ
  - Statistical method: Corpus-based stemming
Porter Stemmer

- It is based on a pattern of vowel-consonant sequence
  - \([C](VC)^m[V]\), \(m\) is an integer

- Rules are divided into steps and examined in sequence
  - Step 1a: ies \(\rightarrow\) i; s \(\rightarrow\); .....  
    cares \(\rightarrow\) care
  - Step 1b: if \(m>0\) eed \(\rightarrow\) ee  
    agreed \(\rightarrow\) agree
  - ..... Step 5a, Step 5b

- Pretty aggressively:
  - nativity \(\rightarrow\) native
K Stemmer: based on morphological rules

- If word occurs in a dictionary, do not stem it

- For all other words
  - Remove inflectional endings: plurals to singular; paste tense to present tense; remove “ing”
  - Remove derivational endings by a sequence of rules: may make mistake when suffixes indicate different meanings like “sign” to “signify”
Text Representation: Word Stemming

Examples of Stemming:

- Original Text:
  Information retrieval deals with the representation, storage, organization of, and access to information items

- Porter Stemmer (Stopwords removed):
  Inform retrieve deal represent storag organ access inform item
Problems with Rule-based Stemming

- Rule-based stemming may be too aggressive e.g., execute/executive, university/universe

- Rule-based stemming may be too conservative e.g., European/Europe, matrices/matrix

- It is difficult to understand the meaning the stems e.g., Iteration/iter, general/gener
Corpus-Based Stemming

- Hypothesis: Word variants that should be considered equally often co-occur in documents (passages or text windows) in the corpus

- Collect the statistics of co-occurrence of words in the corpus and form the connected graph

- Cut the graph by different methods and find the connected subgraphs to form equivalence classes
Text Representation: Word Stemming

(Xu & Croft, 1998)
Text Representation: Process of Indexing

Remove Stopword, Stemming, Phrase Extraction etc

Document → Parser → Text Preprocess

Extract useful fields, useful tokens (lex/yacc)

Indexer

Term Dictionary

Inverted Lists

Document Attributes

Full Text Indexing
Text Representation: Inverted Lists

Inverted lists are one of the most common indexing techniques

- Source file: collection organized by documents
- Inverted list file: collection organized by term
  one record per term, the lists of documents that contain the specific term
- Possible actions with inverted lists
  - OR: the union of lists
  - And: the intersection of lists
## Text Representation: Inverted Lists

### Documents

<table>
<thead>
<tr>
<th>Doc ID</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>kids question noting in 1960s</td>
</tr>
<tr>
<td>2</td>
<td>young man question everything in 1970s</td>
</tr>
<tr>
<td>3</td>
<td>kids question questions in 1980s</td>
</tr>
<tr>
<td>4</td>
<td>young man question nothing in 2000s</td>
</tr>
</tbody>
</table>

### Inverted Lists

<table>
<thead>
<tr>
<th>Term ID</th>
<th>Term</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>kids</td>
<td>1,3</td>
</tr>
<tr>
<td>2</td>
<td>question</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>3</td>
<td>nothing</td>
<td>1,4</td>
</tr>
<tr>
<td>4</td>
<td>in</td>
<td>1,2,3,4</td>
</tr>
<tr>
<td>5</td>
<td>19060s</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>young</td>
<td>2,4</td>
</tr>
<tr>
<td>7</td>
<td>man</td>
<td>2,4</td>
</tr>
<tr>
<td>8</td>
<td>everything</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>1970s</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>questions</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>1980s</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>2000s</td>
<td>4</td>
</tr>
</tbody>
</table>
Many engineering details

- Update inverted lists: delete/insert a term or document
- Compression: trade off between I/O time and CPU time
- Add more information such as position information
- ....................
AD-hoc IR: Basic Process

Information Need

Representation

Query

Retrieval Model

Indexed Objects

Retrieved Objects

Evaluation
Evaluation

Evaluation criteria

- **Effectiveness**
  - How to define effectiveness? Where can we find the correct answers?

- **Efficiency**
  - What about retrieval speed? What about the storage space? Particularly important for large-scale real-world system

- **Usability**
  - What is the most important factor for real user? Is user interface important?
Evaluation

Evaluation criteria

- Effectiveness
  - Favor returned document ranked lists with more relevant documents at the top
  - Objective measures
    - Recall and Precision
    - Mean-average precision
    - Rank based precision

For documents in a subset of a ranked lists, if we know the truth

<table>
<thead>
<tr>
<th></th>
<th>Retrieved</th>
<th>Not retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td>Relevant docs retrieved</td>
<td>Relevant docs not retrieved</td>
</tr>
<tr>
<td>Irrelevant</td>
<td>Irrelevant docs retrieved</td>
<td>Irrelevant docs not retrieved</td>
</tr>
</tbody>
</table>

Precision = \frac{\text{Relevant docs retrieved}}{\text{Retrieved docs}}

Recall = \frac{\text{Relevant docs retrieved}}{\text{ Relevant docs}}
Question: How to find all relevant documents?

Difficult for Web, but possible on controllable corpus

- How to find all relevant documents? (difficult to check one by one)
- Judgers may have inconsistent decisions (subjective judgment)

The Pooling process
Evaluation

Pooling Strategy

- Retrieve documents using multiple methods
- Judge top n documents from each method
- Whole retrieved set is the union of top retrieved documents from all methods
- Problems: the judged relevant documents may not be complete
- It is possible to estimate size of true relevant documents by randomly sampling
Evaluation

System 1

System N
Evaluation

Inconsistent Judgment

- Discussion among multiple judgers to reduce bias
- Combine judgments from multiple judgers
  - Majority vote
- If it is hard to decide for human judgers, it is also hard for automatic system
Evaluation

Evaluate a ranked list

Precision at Recall

- Evaluate at every relevant document

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>0.667</td>
<td>0.2</td>
</tr>
<tr>
<td>0.75</td>
<td>0.3</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4</td>
</tr>
<tr>
<td>0.667</td>
<td>0.4</td>
</tr>
<tr>
<td>0.714</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Not Retrieved: ++++++
Evaluation

Single value metrics

- Mean average precision
  - Calculate precision at each relevant document; average over all precision values

- 11-point interpolated average precision
  - Calculate precision at standard recall points (e.g., 10%, 20%...); smooth the values; estimate 0 % by interpolation
  - Average the results

- Rank based precision
  - Calculate precision at top ranked documents (e.g., 5, 10, 15...)
  - Desirable when users care more for top ranked documents
Evaluation

Sample Results

<table>
<thead>
<tr>
<th>Interpolated Recall (%)</th>
<th>Precision Averages (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>86.49</td>
</tr>
<tr>
<td>10</td>
<td>72.16</td>
</tr>
<tr>
<td>20</td>
<td>64.25</td>
</tr>
<tr>
<td>30</td>
<td>58.40</td>
</tr>
<tr>
<td>40</td>
<td>51.33</td>
</tr>
<tr>
<td>50</td>
<td>44.30</td>
</tr>
<tr>
<td>60</td>
<td>38.43</td>
</tr>
<tr>
<td>70</td>
<td>32.43</td>
</tr>
<tr>
<td>80</td>
<td>21.68</td>
</tr>
<tr>
<td>90</td>
<td>14.40</td>
</tr>
<tr>
<td>100</td>
<td>4.15</td>
</tr>
</tbody>
</table>

Average precision (non-interpolated) for all relevant documents (averaged over queries):

43.06

<table>
<thead>
<tr>
<th>Docs Cutoff Levels</th>
<th>Precision at DCL (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 docs</td>
<td>72.50</td>
</tr>
<tr>
<td>10 docs</td>
<td>67.00</td>
</tr>
<tr>
<td>15 docs</td>
<td>61.83</td>
</tr>
<tr>
<td>20 docs</td>
<td>59.25</td>
</tr>
<tr>
<td>30 docs</td>
<td>55.42</td>
</tr>
<tr>
<td>100 docs</td>
<td>39.75</td>
</tr>
<tr>
<td>200 docs</td>
<td>30.92</td>
</tr>
<tr>
<td>500 docs</td>
<td>19.54</td>
</tr>
<tr>
<td>1000 docs</td>
<td>12.02</td>
</tr>
</tbody>
</table>

R-Precision (precision after R document retrieved, where R = Relevant retrieved):

44.99
Evaluation

TREC collections with queries and relevance judgment

- **TREC CDs 1-5**: 1.5 millions docs, 5GB, news and government reports (e.g., AP, WSJ, Dept of Energy abstracts)
- **TREC WT10g**: crawled from Web (open domain), 1.7 million docs, 10GB
- **TREC Terabyte**: crawled from U.S. government Web pages, 25 million docs, 426 GB
- All have more than 100 queries with relevance judgment
TREC query example

<title> airport security
<desc> Description: What security measures are in effect or are proposed to go into effect in airports?
<narr> Narrative: A relevant document could identify a specific airport and describe the security measures already in effect or proposed for use at that airport. Relevant items could also describe a failure of security that was cited as a contributing cause of a tragedy which came to pass or which was later averted. Comparisons between and among airports based on the effectiveness of the security of each are also relevant.
## Evaluation

### TREC relevance judgment example

<table>
<thead>
<tr>
<th>ID</th>
<th>Document ID</th>
<th>Relevance</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>451</td>
<td>WTX058-B50</td>
<td>85</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B06</td>
<td>411</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B07</td>
<td>154</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B09</td>
<td>203</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B11</td>
<td>245</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B30</td>
<td>262</td>
<td>1</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B37</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B37</td>
<td>149</td>
<td>1</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B37</td>
<td>217</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B37</td>
<td>268</td>
<td>0</td>
</tr>
<tr>
<td>451</td>
<td>WTX059-B37</td>
<td>27</td>
<td>0</td>
</tr>
</tbody>
</table>
Lecture(s) review:

Basic Concepts of Information Retrieval:

- Task Definition of Ad-hoc IR
  - Terminologies and Concepts
  - Overview of Retrieval Models

- Text representation
  - Indexing
  - Text preprocessing

- Evaluation
  - Evaluation methodology
  - Evaluation metrics