Text Retrieval

- Introduction
- Text Retrieval Tasks
- Evaluation
- The Vector Space Model
- Advanced Techniques

Text Retrieval: Overview

- **Goal:**
  - Use a collection of documents to generate a response to a query.
- **Prototypical example:**
  - Web search engine

Readings

- Manning Chapter: Text Retrieval (Selections)
- Vorhees & Harman (Bulkpack)

Text Retrieval Tasks

- **Ad-hoc retrieval**
  - Given: Large fixed document set,
  - Goal: Find documents that answer queries.
- **Filtering**
  - Given: Fixed query
  - Goal: Decide whether documents answer that query.
- **Question-Answering**
  - Given: Large fixed document set
  - Goal: Find short passages (~ 1 sentence) that answer questions.
Text Retrieval Tasks: Classification

There are many different kinds of text retrieval.

- **Variables for Classification:**
  - Is the query known ahead of time?
  - Is the corpus known ahead of time?
  - What kind of query is used?
  - What kind of corpus is used?
  - What kind of response should be generated?

---

Text Retrieval Tasks: Classification (Cont'd)

- **Example classifications:**

<table>
<thead>
<tr>
<th>Task</th>
<th>Known ahead of time?</th>
<th>Corpus</th>
<th>Query</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-Hoc Document</td>
<td>Corpus</td>
<td>Short – Medium</td>
<td>Document</td>
<td></td>
</tr>
<tr>
<td>Question Answering</td>
<td>Corpus</td>
<td>Short</td>
<td>Sentence</td>
<td></td>
</tr>
<tr>
<td>Filtering</td>
<td>Query</td>
<td>Long</td>
<td>Document</td>
<td></td>
</tr>
<tr>
<td>Cross-Language</td>
<td>Corpus</td>
<td>Multilingual</td>
<td>Short</td>
<td>Document</td>
</tr>
<tr>
<td>Speech</td>
<td>Corpus</td>
<td>Speech</td>
<td>Short – Medium</td>
<td>Document</td>
</tr>
</tbody>
</table>

---

TREC

- **Text REtieval Conference**
  - Annual workshop to foster research in text retrieval
  - 50+ groups compete for high performance.
- **16 Tracks = types of text retrieval**
  - 7 tracks were active in TREC-9

<table>
<thead>
<tr>
<th>Track</th>
<th>Database Merging</th>
<th>High Precision</th>
<th>Very Large Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad-Hoc</td>
<td>Filtering</td>
<td>Chinese</td>
<td>Query</td>
</tr>
<tr>
<td>Routine</td>
<td>NLP</td>
<td>Cross-Language</td>
<td>View</td>
</tr>
<tr>
<td>Interactive</td>
<td>Speech</td>
<td>Web</td>
<td></td>
</tr>
</tbody>
</table>

---

Evaluation

- **Precision/Recall: Review**

<table>
<thead>
<tr>
<th>Target</th>
<th>~Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected</td>
<td>True positive</td>
</tr>
<tr>
<td>~Selected</td>
<td>False negative</td>
</tr>
</tbody>
</table>

- **Precision = \( \frac{tp}{tp + fp} \)**
  - What proportion of selected items are correct?

- **Recall = \( \frac{tp}{tp + fn} \)**
  - What proportion of target items are selected?
Evaluation: Precision/Recall Graphs

- Text retrieval generates multiple responses
- Consider the first $n$ responses for various $n$
  - $n=0$: precision=100%; recall=0%
  - $n=N$: precision=0%; recall=100%
- Graph precision vs. recall at various cutoffs

Evaluation: Summary Measures

- Uninterpolated average precision
  - $P = \text{positions at which we got a true positive}$
  - average precision = $\frac{\text{Avg(precision at cutoff } p)}{p \in P}$
- F Measure
  - Weighted harmonic mean of precision and recall.
  $$\frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

Evaluation: Test Set Construction

- Which documents are relevant?
  - Need human evaluation
  - New evaluation for each query
- Need a complete set of relevant documents
  - Can't find recall without a complete set
- But there are too many documents to evaluate!

Evaluation: Pooling

- TREC: Construct a test set that is "approximately correct"
  - Assume that all relevant documents are returned by at least one text retrieval system
- Manually evaluate the top 100 documents returned by each system.
- No bias against any competing system.
- Small bias against systems that did not compete.
The Vector Space Model

Represent each document as a sparse vector $\mathbf{x}$

- Each dimension corresponds to a single vocabulary item:
  - 1 if the vocabulary item is in the document
  - 0 if the vocabulary item is not in the document

\[ d = \text{document (multiset of words)} \quad \mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle \]
\[ V = w_1, w_2, \ldots, w_n = \text{vocabulary} \]
\[ w_i = \text{word} \]

Comparing Documents

- Two documents are similar if they have similar term vectors.

Comparing Documents & Queries

- Treat the query as a very small document.
- Construct a vector representation of the query.

- Query vectors are typically shorter than document vectors.

Comparing Term Vectors

What makes term vectors similar?

- **Attempt 1:** Term vectors are similar if their difference is small.

  \[ \text{similarity}_1(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2| \]

- Does document length matter?
- We only care about the relative frequency of each term.
Comparing Term Vectors Cont'd: The Cosine Measure

What makes term vectors similar?

- **Attempt 2**: Term vectors are similar if the angle between them is small.

\[
similarity_2(x^*_1, x^*_2) = \cos(x^*_1, x^*_2) = \frac{x^*_1 \cdot x^*_2}{|x^*_1||x^*_2|}
\]

Normalizing Term Vectors

- Cosines are easier to compute if we first normalize all document vectors.

\[
x' = \frac{x}{\|x\|} = \frac{x}{\sqrt{\sum_{i=1}^{n} x_i^2}}
\]

\[
similarity_2(x^*_1, x^*_2) = x^*_1 \cdot x^*_2
\]

Term Weighting

- Some terms are more informative than others.

- Can we get a better similarity measure if we use different vectors?  
  \[x = (x_1, x_2, ..., x_n)\]

- Change our definition of \(x_i\)

  \[x_i = \begin{cases} 
  1 & \text{if } w_i \in d \\
  0 & \text{if } w_i \notin d 
  \end{cases}
  \]

- Increase \(x_i\) for more informative terms.

Term Weighting (Continued) Term Frequency

- **Term frequency**: \(tf_i = \text{number of occurrences of } w_i \in d\)

- Higher term frequency ➔ term is more relevant to the document.

- Weight a term proportionally to its term frequency? \(x_i = tf_i\)

- But a word appearing 3 times as often is not 3 times as relevant: use smoothing.

  \[x_i = \begin{cases} 
  1 + \log(tf_i) & \text{if } w_i \in d \\
  0 & \text{if } w_i \notin d 
  \end{cases}
  \]

  or \(x_i = \sqrt{tf_i}\)
Term Weighting (Continued)
Inverse Document Frequency

- **Document Frequency:**
  \[ df_i = \text{number of documents containing } w_i \]
- Higher document frequency ➔ term is less informative.
- Weight a term inversely to its document frequency.
  \[ x_i = \begin{cases} 
  \left(1 + \log(tf_i)\right) \log\left(\frac{N}{df_i}\right) & : \text{if } w_i \in d \\
  0 & : \text{if } w_i \notin d 
  \end{cases} \]
- (use smoothing)

Latent Semantic Indexing

- Vector model assumes that term contributions to document similarity are independant.
- Many terms are not independant.
  - "star chart" is similar to "astral map"
- Project document vectors into a new vector space.
  - In the new vector space, terms that are semantically similar are close to each other.
  - "Latent" semantic dimensions

Using NLP Techniques

- **Traditional wisdom: linguistic knowledge does not improve text retrieval performance.**
  - Especially true for ad-hoc retrieval.
- **But:**
  - Linguistic knowledge useful for other tasks (e.g., question answering, cross-language text retrieval)
  - Clearly a temporary fact: it is impossible to get 100% performance without linguistic knowledge.

Using NLP Techniques: Words

- **Add new terms:**
  - Find terms with collocation analysis
  - Synonym expansion: WordNet
- **Make terms more precise:**
  - Word sense disambiguation
- **Merge terms with the same meaning:**
  - Stemming
- **Clean up a "messy" corpus:**
  - Language modelling
  - Spell-check
Using NLP Techniques:
Syntax

• **Find new terms:**
  - Entity extraction
  - Relationship extraction

• **Narrow the search space:**
  - Question parsing
    • Determine what type of answer we expect
  - Entity tagging
    • Tag entities with types (e.g., company, person, date)
  - Relationship extraction
    • Search for partial relationships (e.g., "John likes ?x")