Some slides in this presentation are borrowed from Chris Manning and Dan Jurafsky

Last time

• Information Retrieval:
  – A collection of documents, A query
  – Find the relevant document set.

• Build an inverted index:
  – Term ➔ \{(doc1, f1), (doc2, f2), (doc3, f3), \ldots\}

• Boolean search
  – Optimization of queries
Problem with Boolean search: feast or famine

- Boolean queries often result in either too few (≈0) or too many (1000s) results.
  - Query 1: “standard user dlink 650” → 200,000 hits
  - Query 2: “standard user dlink 650 no card found” → 0 hits
- It takes a lot of skill to come up with a query that produces a manageable number of hits.
  - AND gives too few; OR gives too many

Ranked retrieval models

- Rather than a set of documents satisfying a query expression, in ranked retrieval models, the system returns an ordering over the (top) documents in the collection with respect to a query
- Free text queries: Rather than a query language of operators and expressions, the user’s query is just one or more words in a human language
Feast or famine: not a problem in ranked retrieval

- When a system produces a ranked result set, large result sets are not an issue
  - Indeed, the size of the result set is not an issue
  - We just show the top $k$ ($\approx 10$) results
  - We don’t overwhelm the user

- Premise: the ranking algorithm works

Vector Space Model

- Documents and queries are represented as vectors:
  - Each dimension represents a term in the collection

$$\vec{d}_j = (t_{1j}, t_{2j}, \ldots, t_{Nj})$$
$$\vec{q}_k = (t_{1k}, t_{2k}, \ldots, t_{Nk})$$

- $d_j$ for the $j$th document and $q_k$ for a query ($k$).
Term Frequency (TF)

• We use weights to represent the importance of a given term in document (or query).

\[
\tilde{d}_j = (w_{1j}, w_{2j}, \ldots, w_{Nj})
\]

\[
\tilde{q}_k = (w_{1k}, w_{2k}, \ldots, w_{Nk})
\]

— First estimation: Frequency of the term in a document (or query)

Remember: Constructing an index

• **Start:** words and their corresponding document

• **End:** Inverted index with frequencies
Remember: Constructing an index

Example: Vector representation

- Doc-2: Ali cut apple and prepared apple juice.
- Doc-3: Tariq ate at home with Ali.

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Mona</th>
<th>eat</th>
<th>apple</th>
<th>home</th>
<th>prepared</th>
<th>juice</th>
<th>Tariq</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D1</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>D2</strong></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>D3</strong></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Similarity in the vector space

\[
\vec{d}_j = (w_{1j}, w_{2j}, \ldots, w_{Nj})
\]
\[
\vec{q}_k = (w_{1k}, w_{2k}, \ldots, w_{Nk})
\]

- **Intuition:**
  Vectors (documents) which are close talk about similar things.

Reviewing vector space operation

- **Dimension:** number of elements
  - Size of vocabulary
- **Length:**
  \[
  |X| = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2} = \sqrt{\sum_{i=1}^{n} x_i^2}
  \]
Reviewing vector space operation

- Dimension: number of elements
  - Size of vocabulary
- Length:
  \[ |\bar{X}| = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2} = \sqrt{\sum_{i=1}^{n} x_i^2} \]
  - Example: \( \bar{d} = (3, 4) \)
  \[ |\bar{d}| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5 \]

Cosine Similarity

- Similarity between two vectors
  - Cosine of the angle between the two vectors
\[
\cos(x, y) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}
\]
Interpreting cosine similarity

\[ \cos(x, y) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \]

- Two vectors (documents) with the same elements (terms) but different values get high similarity.

\[ \cos((4, 3, 0), (3, 4, 0)) = \frac{(4, 3, 0) \cdot (3, 4, 0)}{\sqrt{10+9} \sqrt{9+10}} \]

Similarity function

- Cosine similarity for a document and a query.

\[ \cos(x, y) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \]

\[ \text{sim}(d_j, q_h) = \frac{d_j \cdot q_h}{|d_j||q_h|} = \frac{\sum_{i=1}^{n} w_{ij} w_{hk}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2} \sqrt{\sum_{i=1}^{n} w_{hk}^2}} \]
Beyond term frequency

- We use weights to represent the importance of a term
  - First estimation: term frequency


<table>
<thead>
<tr>
<th>Ali</th>
<th>Mona</th>
<th>eat</th>
<th>apple</th>
<th>home</th>
<th>prepared</th>
<th>juice</th>
<th>Tariq</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Is term frequency sufficient?

IDF example

<table>
<thead>
<tr>
<th>Ali</th>
<th>Mona</th>
<th>eat</th>
<th>apple</th>
<th>home</th>
<th>prepared</th>
<th>juice</th>
<th>Tariq</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Ali: present in all docs
- Tariq: only in one doc
Inverse Document Frequency (IDF)

• An infrequent term can be important
  – When it happens in small number of documents.
• IDF:
  \[ idf_t = \log \frac{N}{n_t} \]
  – \( N \): number of documents in the collection
  – \( n_t \): number of documents that \( t \) occurs.
• If a term occurs in all documents, its IDF is zero.

IDF example

<table>
<thead>
<tr>
<th></th>
<th>Ali</th>
<th>Mona</th>
<th>eat</th>
<th>apple</th>
<th>home</th>
<th>prepared</th>
<th>juice</th>
<th>Tariq</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

• Ali: present in all docs → \( idf = 0 \)
• Tariq: only in one doc
TF-IDF

• Instead of using the term frequency as the weight, we use a combination of two factors

\[ W_{t,d} = tf_{t,d} \times idf_t \]

\[ \text{sim}(d_j, q_k) = \frac{d_j \cdot q_k}{|d_j| |q_k|} = \frac{\sum_{i=1}^{n} w_{ij} \cdot w_{ik}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2} \sqrt{\sum_{i=1}^{n} w_{ik}^2}} \]

Handling unseen terms

• Unseen terms \( \Rightarrow \) zero division in IDF
• Simplification for unseen terms:
  – If query is all unseen terms \( \Rightarrow \text{sim}(d_j, q_k) = 0 \)
  – For individual unseen terms \( \Rightarrow W_{t,d} = 0 \)
Log weighing of TF-IDF

• Few variants of TF-IDF
  – Using log weighing of IDF
    – If $tf > 0$ \( w_{r,d} = (1 + \log tf_{r,d}) \times \log_{10}(N / df_r) \)
    – If $tf = 0$ \( W_{r,d} = 0 \)

Effect of idf on ranking

• Question: Does idf have an effect on ranking for one-term queries, like
  – iPhone
Effect of idf on ranking

- Question: Does idf have an effect on ranking for one-term queries, like
  - iPhone
- idf has no effect on ranking one term queries
  - idf affects the ranking of documents for queries with at least two terms
  - For the query capricious person, idf weighting makes occurrences of capricious count for much more in the final document ranking than occurrences of person.

Collection vs. Document frequency

- The collection frequency of $t$ is the number of occurrences of $t$ in the collection, counting multiple occurrences.
- Example:

<table>
<thead>
<tr>
<th>Word</th>
<th>Collection frequency</th>
<th>Document frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>insurance</td>
<td>10440</td>
<td>3997</td>
</tr>
<tr>
<td>try</td>
<td>10422</td>
<td>8760</td>
</tr>
</tbody>
</table>

- Which word is a better search term (and should get a higher weight)?
TF-IDF Ranking

The underlying framework for most search systems.

\[
sim(d_j, q_h) = \frac{d_j \cdot q_h}{|d_j||q_h|} = \frac{\sum_{t=1}^{N} w_{t,j} w_{t,h}}{\sqrt{\sum_{t=1}^{N} w_{t,j}^2 \cdot \sqrt{\sum_{t=1}^{N} w_{t,h}^2}}}
\]

\[
W_{t,d} = tf_{t,d} \cdot idf_t
\]

Summary – vector space ranking

- Represent the query as a weighted tf-idf vector
- Represent each document as a weighted tf-idf vector
- Compute the cosine similarity score for the query vector and each document vector
- Rank documents with respect to the query by score
- Return the top \( K \) (e.g., \( K = 10 \)) to the user
Estimating relevance via classification

• Going beyond the bag of word information
• Benefit from the documents’ info
  – Specially on the web

Integrating multiple features to determine relevance

• Modern systems – especially on the Web – use a great number of features:
  • Arbitrary useful features – not a single unified model
  – Log frequency of query word in anchor text?
  – Query word in color on page?
  – # of images on page?
  – # of (out) links on page?
  – URL length?
  – URL contains “~”?
  – Page edit recency?
  – Page length?
• The New York Times (2008-06-03) quoted Amit Singhal as saying Google was using over 200 such features.
How to combine features to assign a relevance score to a document?

• Given lots of relevant features...
• you can build a classifier to learn weights for the features
  – Requires: labeled training data
  – This is the “learning to rank” approach, which has become a hot area in recent years
    • I only provide an elementary introduction here

Simple example:
Using classification for ad hoc IR

• Collect a training corpus of \((q, d, r)\) triples
  – Relevance \(r\) is here binary (but may be multiclass, with 3–7 values)
  – Train a machine learning model to predict the class \(r\) of a document-query pair
    • ... just like when we were doing text classification
IR evaluation

• IR as a machine learning problem:
  – Training: collect the frequencies, build an index
  – Testing: Search

• Testing Scenario:
  – A collection of N documents
  – A collection of Q queries
    • A set of relevant documents labeled by humans

Confusion Matrix

• Breaking down the errors

<table>
<thead>
<tr>
<th>IR system</th>
<th>Human Judgments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td>Relevant</td>
</tr>
<tr>
<td>Not Relevant</td>
<td>Not Relevant</td>
</tr>
</tbody>
</table>

- True Positive (TP)
- False Positive (FP)
- False Negative (FN)
- True Negative (TN)
Basic Evaluation measures

- **Accuracy**: Percentage of documents correctly classified by the system
  \[ \text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \]

- **Error Rate**: Percentage of documents incorrectly classified by the system
  \[ \text{Error Rate} = \frac{FP + FN}{TP + FP + TN + FN} \]

Accuracy?

- **Is accuracy a good measure for IR?**
Accuracy?

• Is accuracy a good measure for IR?
  – Number of true negatives (TN)?

• Usually high: billions
  – True positives, false positives, false negatives
    • Low: hundred

• Need a measure that doesn’t rely on TN.
Precision

• Precision: What percentage of the returned documents are relevant?
  – How precise does the system work?

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

Recall

• Recall: What percentage of the gold standard relevant document, does the IR system finds?
  – How is the coverage of the system?

\[
\text{Recall} = \frac{TP}{TP + FN}
\]
The F1 measure

• Combining Precision and Recall into one global metric.

\[ F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

• Used beyond IR:
  – Document classification
  – Named entity recognition

Mean Average Precision

• **Problem**: Precision and Recall measure don’t care about the order (ranking)
  – Web search: 1st page matters a lot

• Mean Average Precision: At a given rank (e.g 10), what is the precision?
Web Search

Web search basics

User

Web spider

Indexer

The Web

Ad indexes

Carnegie Mellon University in Qatar

Ad indexes

User

Web spider

Indexer

15383: txt proc

The Web

Ad indexes

User

Web spider

Indexer

15383: txt proc
Web crawler

• Need to collect as many documents to build a rich index
  – Web crawlers: text robots which walk through the web and collect documents.

• Crawling scenario:
  – Start with seed URLs, parse the document to find new links
Web crawler

- Need to collect as many documents to build a very rich index
  - Web crawlers: text robots which walk through the web and collect documents.

- Crawling scenario:
  - Start with seed URLs, parse the document to find new links
  - Expand the search space with every link, while maintaining a central list repository.
  - Build and update the web graph

A Web graph

- A graphical representation of the web
A Web graph

• Similarity among query and document is not enough for Web search!

• Take advantage of hyperlinks
  – Popularity of a webpage

• Page Rank

PageRank: is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.
Page Ranking algorithm

• Similarity among query and document is not enough for Web search!
  Take advantage of hyperlinks!

• PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.

Search and HCI

• Human computer interaction influences search strategies
  – Specially on the web
How far do people look for results?

“When you perform a search on a search engine and don’t find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)”

- 12% After reviewing the first few entries
- 16% After reviewing the first page
- 26% After reviewing the first 2 pages
- 27% After reviewing the first 3 pages
- 20% After reviewing more than 3 pages

(Source: iprospect.com WhitePaper_2006_SearchEngineUserBehavior.pdf)

Relevance Feedback

- **Problem:** Initial query might not be sufficient
- **Idea:** Modify the query to retrieve more relevant documents
- **Involves interaction with the user**
How to expand the query

- Use Thesaurus and query logs to expand
  - Book: publication, author, composition
  - Newspaper: press, news, newsprint

Pseudo Relevant Feedback

- Run the query and collect the results
- Extract N new frequent terms from the top M documents.
Pseudo Relevant Feedback

- Run the query and collect the results
- Extract N new frequent terms from the top M documents.
- Expand the terms to the original query and run the new query
- Provide the results

The trouble with paid search ads ...

- It costs money. What’s the alternative?
- Search Engine Optimization:
  - “Tuning” your web page to rank highly in the algorithmic search results for select keywords
    - Alternative to paying for placement
    - Thus, intrinsically a marketing function
The trouble with paid search ads ...

- It costs money. What’s the alternative?
- *Search Engine Optimization:*
  - “Tuning” your web page to rank highly in the algorithmic search results for select keywords
    - Alternative to paying for placement
    - Thus, intrinsically a marketing function
- Performed by companies, webmasters and consultants (“Search engine optimizers” -- SEOs) for their clients
- Some perfectly legitimate, some very shady

Simplest forms

- First generation engines relied heavily on *tf-idf*
  - The top-ranked pages for the query `maui resort` were the ones containing the most `maui`’s and `resort`’s
- SEOs responded with dense repetitions of chosen terms
  - e.g., `maui resort maui resort maui resort`
  - Often, the repetitions would be in the same color as the background of the web page
    - Repeated terms got indexed by crawlers
    - But not visible to humans on browsers

Pure word density cannot be trusted as an IR signal
Adversarial IR

- Search engines have responded to this in many ways:
  - Quality/spam detection measures on pages
  - Use of other metrics such as link analysis, user votes
- But it’s a fundamentally new world:
  - Before, we assumed that the documents just existed independently, and we could build an IR system for them
  - Now, the documents are being changed in ways that attempt to maximize their ranking in search results

Adversarial IR

- Search engines have responded to this in many ways:
  - Quality/spam detection measures on pages
  - Use of other metrics such as link analysis, user votes
- But it’s a fundamentally new world:
  - Before, we assumed that the documents just existed independently, and we could build an IR system for them
  - Now, the documents are being changed in ways that attempt to maximize their ranking in search results
- **Adversarial IR:** the unending (technical) battle between SEO’s and web search engines
  - For more see: [http://airweb.cse.lehigh.edu/](http://airweb.cse.lehigh.edu/)