Indexing & retrieval

Example “Collection”

S1 Tropical fish include fish found in tropical environments around the world, including both freshwater and saltwater species.

S2 Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

S3 Tropical fish are popular aquarium fish, due to their often bright coloration.

S4 In freshwater fish, this coloration typically derives from iridescence, while saltwater fish are generally pigmented.

Four sentences from the Wikipedia entry for tropical fish

Announcements

• Assignment 2
  – Due Wednesday, 11:59PM

• Office hours changes
  – My normal office hours tomorrow (2-3pm) cancelled
  – Today 5:20pm
  – Tomorrow 5:00pm

Simple Inverted Index

Handling phrases

Inverted index with counts

• Records # of times word appears in each document
• Supports better ranking algorithms
Phrases

• How do we index phrases?
• Three possible approaches:
  – Identify syntactic phrases using a part-of-speech (POS) tagger
  – Use word n-grams
  – Store word positions in indexes

Part-of-speech Tagging

• One approach: use Natural Language Processing to parse a document into phrases
• Part-of-speech (POS) taggers use statistical models of text to predict syntactic tags of words
  – Example tags:
    • NN (singular noun), NNS (plural noun), VB (verb), VBD (verb, past tense), VBN (verb, past participle), JJ (adjective)...
    – Phrases can then be defined as simple noun groups, for example

Pos Tagging Example

Original text:
Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of each chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulus demand, price or volume of sales.

Tagging:
Documents/NW will describe/N agrochemicals/N market share/N predicted/N stimulus/N demand/N price/N sales/N.
N-Grams

- Frequent n-grams are more likely to be meaningful phrases
- N-grams form a Zipf distribution
- Could index all n-grams up to specified length
  - Uses a lot of storage
    - e.g., document containing 1,000 words would contain 3,990 instances of word n-grams of length $2 \leq n \leq 5$

Google N-Grams

- Many web search engines index n-grams
- Google sample:
  - Number of tokens: 1,024,988,272
  - Number of sentences: 95,119,465
  - Number of unigrams: 13,588,391
  - Number of bigrams: 31,843,401
  - Number of trigrams: 97,069,902
  - Number of fourgrams: 1,313,918,354
  - Number of fivegrams: 1,176,470,663
- Most frequent trigram on English web is... ?

“Culturomics”

- Emerging field using digital tools to study arts, history, and literature
  - Example: Harvard/Google books n-grams database
  - http://ngrams.googlelabs.com/

Google Books n-grams

http://ngrams.googlelabs.com/

Google Books n-grams

http://ngrams.googlelabs.com/

Google Books n-grams

http://ngrams.googlelabs.com/
Phrases

• How do we index phrases?
• Three possible approaches:
  – Identify syntactic phrases using a part-of-speech (POS) tagger
  – Use word n-grams
  – Store word positions in indexes and use proximity operators in queries

Inverted index with positions

• supports proximity matches
Proximity Matches

- Matching phrases
  - e.g., "tropical fish"
- Word position information in inverted lists make these types of query features efficient
  - e.g.,

<table>
<thead>
<tr>
<th>word</th>
<th>1.1</th>
<th>1.4</th>
<th>2.0</th>
<th>2.7</th>
<th>2.1</th>
<th>5.1</th>
<th>5.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>tropical fish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Advanced features

- Index compression
- Distributed indexing
- Distributed retrieval
- Caching

Compression

- Inverted lists are very large
  - e.g., 25-50% of collection for TREC collections using Indri search engine
  - Much higher if n-grams are indexed
- Compression of indexes saves disk and/or memory space
  - Typically have to decompress lists to use them
  - Best compression techniques have good compression ratios and are easy to decompress

Distributed Indexing

- Indexing huge amounts of data is computationally expensive
  - Large numbers of inexpensive servers used rather than larger, more expensive machines
  - MapReduce is a distributed programming tool designed for indexing and analysis tasks

Distributed Evaluation

- Basic process
  - All queries sent to a director machine
  - Director then sends messages to many index servers
  - Each server does some portion of the query processing
  - Director organizes results and returns them to the user
- Two main approaches
  - Document distribution (most popular)
  - Term distribution

Distributed Evaluation

- Document distribution
  - each index server acts as a search engine for a small fraction of the total collection
  - director sends copy of the query to each index server, each of which returns the top-k results
  - results are merged into a single list by the director
Caching

• Query distributions similar to Zipf
  – About ½ of queries each day are unique, but others are very popular

• Caching can significantly improve effectiveness
  – Cache popular query results
  – Cache common inverted lists
  – Must be refreshed to prevent stale data