COMP38120 - Documents, Services and Data on the Web

Sam Littlefair

October 2, 2018

1 Course Unit Structure

- 20 credit module with 1 exam in Summer worth 60%.
- 2 labs due in Semester 2:
  1. Week 3 - 25%.
  2. Week 11 - 15%.

2 Documents on the Web

The "Traditional" Web: Web pages, images, videos, audio files, etc. Collectively known as documents - objects that can be indexed (so that it can be found) and searched for.

2.1 Information retrieval

Information retrieval is finding material of an unstructured nature that satisfies an information need from within large collections (Manning et al 2008). IR is not just search engines, it can be other things such as voice controlled assistants like Siri, or music recognition like Shazam. There is no obvious structure, unlike a rigidly structured database - but doesn't mean there's no structure at all, there is document structure (paragraphs, headings..) and mark-up (XML, HTML).

2.2 Informational need

Informational need is the topic about which the user desires to know more about.

- **Informational:** Need to learn about something (80% of queries)
- **Navigational:** Need to go to some page (10%)
- **Transactional:** Need to do something using the Web (10%)

How do we distinguish between them?

Navigational

- Company/business/org name
- Domain suffix
- Length of query (<3)

Transactional

- Queries with 'interact' terms (buy, chat,...) or 'obtaining' terms (e.g. 'download', 'get')
- Terms related to movies, songs, lyrics, recipes, images, humour
  - Image, audio or video collections ('audio', 'images', 'video')
  - Movies, songs, images, multimedia, compression file extensions (jpg, zip, ...)
  - Entertainment terms (pictures, games, ...)

Informational

- Use of question words, many phrases
- Informational items ('list of...')
- Query length >2
- Neither navigational nor transactional i.e., easier to identify N and T (?)

2.3 User Behaviour

- About 3 searches per day for an average user.
- 16% to 20% of queries that get asked every day have never been asked before.
- 78% of queries not modified.
- User queries are very short. 38% of queries single word, 22% 2 words etc. All full of spelling mistakes!

2.4 Information Retrieval System

An IR system finds the similarities of a query representation to document representation. Documents must be indexed – each document is reduced to set of index terms.

- Index language: A means of representing documents and queries.
- Search is focused on index: Map queries to index i.e. determine similarity of a query to a document via index items.

2.4.1 Indexing process (offline)

1. Acquire documents through crawling, feeds, etc.
2. Generate index for each document
3. Store the index in an easy-to-search form

2.4.2 The retrieval process (online)

1. Assist the user in formulating the query (did you mean, autocomplete..)
2. Transform the query ("index" the query)
3. Map the query's representation against the index
4. Retrieve and rank the results
5. Help users browse the results and re-formulate the query
6. Log user's actions etc.

2.5 Representation

2.5.1 PubMed - Medical Publications

MeSH (Medical Subject Headings) is the a manually controlled vocabulary used for indexing articles for PubMed. Very hard to maintain, indexing can take months.

2.5.2 Bag of Words (BoW)

An indexing model which reduces the document into a map of words and their frequencies.

Pros

- IR matching is about the degree of overlap between a query and document representations so it's efficient.

Cons

- Meaning could be lost without order, but not a huge deal.
- Meaning could be lost without context.
- Might not work for all languages.
- Negations are lost.

2.5.3 Term-document Matrix

Store the number of occurrences in a matrix where one dimension is terms, and the other is the documents.

$$\begin{array}{c} \\ \textit{complexity} \\ \textit{algorithm} \\ \textit{entropy} \end{array} \begin{array}{c} \begin{array}{ccc} \textit{Doc}1 & \textit{Doc}2 & \textit{Doc}3 \end{array} \\ \begin{pmatrix} 4 & 0 & 3 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

2.5.4 Term-document Incidence Matrix

Just store a 1 if it occurs, 0 otherwise.

$$\begin{array}{c} \\ \textit{complexity} \\ \textit{algorithm} \\ \textit{entropy} \end{array} \begin{array}{c} \begin{array}{ccc} \textit{Doc}1 & \textit{Doc}2 & \textit{Doc}3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

The problem with matrix approach is the matrices will be massive. With 1 million documents, 1000 words per document, and 500k unique words you'd have a matrix of size 500 billion.

2.5.5 Inverted Index

For each term, store a list of all documents that contain it.
The document list for each term will be of varying length so we can use linked lists or variable sized arrays.

Brutus → 1 2 4 11 31 45 173 174

Caesar → 1 2 4 5 6 16 57 132

Calpurnia → 1 2 2 31 54 101

Indexer steps

1. **Token sequence**
   - Create a sequence of Token, Document ID pairs.
2. **Sort**
   - Sort by terms, then by document ID.
3. **Dictionary & Postings**
   - Multiple term entries in a single document are merged
   - Split into Dictionary and Postings
   - Document frequency (for each term) is also added