Multimedia Retrieval Notes

Lecture 1: Overview of Multimedia Data Management

Multimedia

- Multimedia: any combination of two or more media forms and well-integrated to be presented via a single interface, or manipulated by computer program
- Enriches the experiences in obtaining information
- Multimedia applications are changing our life → education, healthcare, entertainment, research
- **Issues related to multimedia**
    - Multimedia development
    - Multimedia storage
    - Multimedia retrieval
    - Multimedia delivery (exchange)

Multimedia development

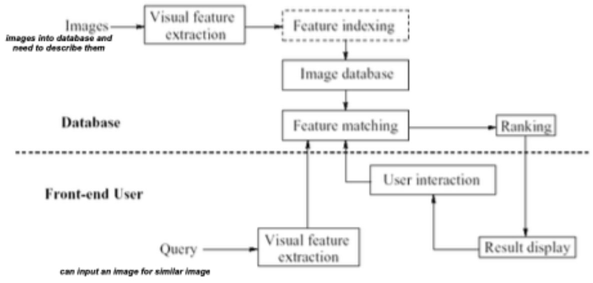
- **Concept definition →** program goals, outline, logic flow chart
- **Storyboard design →** represents the proposed multimedia project graphically, with a series of templates
- **Development of multimedia elements →** such as text, graphics, video, sound, animation are produced using a variety of software applications and imported into the multimedia authoring systems
- **Authoring →** authoring systems assembly all the multimedia elements and then import and integrate them into a comprehensive and possible interactive application
- **Testing and revision →** testing, evaluating and revising your product
- **Delivery →** distributing the final multimedia production
- **Data deluge -** The data deluge refers to the situation where the sheer volume of new data being generated is overwhelming capacity of institutions to manage it and researchers to make use of it

Multimedia storage

- Digitisation: to be manipulated by computer
- Compression: to save space
- Storage media: where to store and how to manage

Multimedia Retrieval

- Feature extraction is performed to obtain multi-dimensional feature vectors characterising multimedia contents
- These features are indexed
- Appropriate similarity measurement is employed to measure the similarity between query item and database item
- Feedback provides the interactions between users and systems
- Efficient indexing techniques are employed to organise databases



Images
images into database and need to describe them
Visual feature extraction
Feature indexing
Image database
Database
Feature matching
Ranking
Front-end User
User interaction
Query
can input an image for similar image
Visual feature extraction
Result display

Lecture 2: Text information retrieval

Topic summary

Document Representation

For each term in a document, we store a set of all documents that contain t. This can be stored in a **term-document incidence matrix.** However, this matrix, whilst easy, is not very efficient as it has a lot of cells. The good news is that the matrix is very sparse. Thus, we can create what we call an **inverted file.** In this structure, we have a dictionary containing the terms, and postings, containing which documents contain this term.

Whilst boolean retrieval is accurate for precise information needs, it is not flexible. Something is either a match or it isn't. We need to be able to consider position information. That is, information about phrases and the proximity of words to each other. We also should be able to rank documents based on their relevance. We can use **term-document count matrices** to combat this. Here, each document is a count vector of how many times terms appeared in the document. However, this model using vector representation is called the bag of words model and doesn't consider the ordering of words in a document. The **term frequency** of a term in a document is the number of times that term occurs in the document. However, we don't care about this raw frequency. A document with 10 occurrence of the term is not 10 times more relevant than a document with one occurrence. Thus, we use **log-frequency weighting** to counteract this. Rare terms are more informative than frequent terms. We want a higher weight for rare terms. For frequent terms, we want positive weights for common words but lower weights than for rare terms. We use **document frequency,** the number of documents that contain the term, to capture this score. **Idf,** the inverse of document frequency is used to measure the informativeness of t. **Collection frequency** is the number of occurrences of the term in the collection. **tf-idf weight** of a term is the product of its tf weights with its idf weight and is the best known weighting scheme in information retrieval as it increases with the number of occurences within a document and with the rarity of the term within the collection. Each document can now be represented by a vector of tf-idf weights – one for each word. We can now represent queries as vectors in the space also and rank the documents according to their proximity to the query in space. We don't want to use Euclidean distance to do this as it is very large for vectors of different lengths. Instead of distance, we use angles. We rank documents in decreasing order of the angle between the query and document. Thus, we rank documents in increasing order of cosine(query, document). We can normalise the length of a vector by dividing each of its components by its length.

Relevance feedback

The user gives feedback on the relevance of documents in the initial set of results by marking some results as relevant or not relevant. The system computes a better representation of the information need based on feedback.

A key concept in this is a **centroid.** The centroid is the centre of mass of a set of points. As we represent documents as points in a high-dimensional space, we can make use of this. The **Rocchio Algorithm** uses the vector space model to pick a relevance fed back query. Three associated weights are used to shape the modified vector in a direction closer or farther away from the original query, related documents and non-related documents. Relevance feedback can improve recall and precision.

However, there are some assumptions we are making with relevance feedback. Firstly, we assume that the user has sufficient knowledge for the initial query. However, the user might not have sufficient knowledge e.g. misspellings. Second, we assume relevance prototypes are 'well behaved'. However, sometimes there can be several relevance prototypes e.g. Burma/Myanmar. There are other problems with relevance

feedback. Long queries are inefficient for types of information retrieval systems, causing long response times and high cost to the retrieval systems. Users are often reluctant to provide explicit feedback.
We can also use a technique called **query expansion.** In query expansion, users give additional input on words or phrases. We can augment the user query through a manual thesaurus or global analysis such as using an automatically derived thesaurus, refinements based on query log mining and local analysis of documents in a result set. In **thesaurus based query expansion,** for each term we expand the query with synonyms and related words to the term from the thesaurus. We may weight added terms less than original query terms. In **automatic thesaurus generation,** we attempt to generate a thesaurus automatically by analysing the collection of documents. We denote two words as similar if they co-occur with similar words. We denote two words as similar if they occur in a given grammatical relation with the same words. This is called a **co-occurrence thesaurus.**

Result summaries

Having ranked the documents matching a query, we wish to present a results list. Most commonly, a list of all the document titles plus a short summary. A **static summary** of a document is always the same, regardless of the query that hit the document. A **dynamic summary** is a query dependent attempt to explain why the document was retrieved for the query at hand.

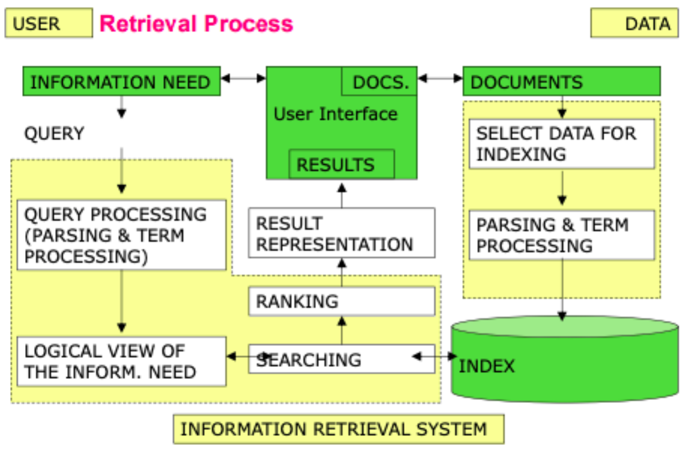
Evaluating an IR system

Precision is the fraction of retrieved documents that are relevant. **Recall** is the fraction of relevant documents that are retrieved. A combined measure that assesses the precision/recall trade-off is the F measure. We can also use **A/B testing.**

*****

Background

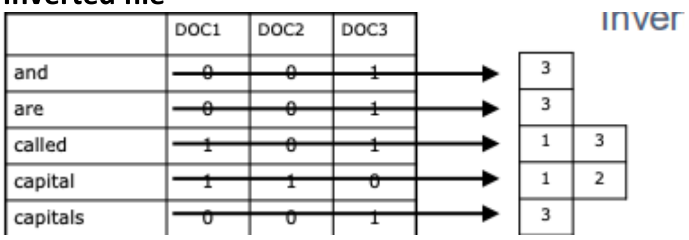
- There is far more unstructured text out there than structured text
- About 30 years ago, more and more alphanumeric data was stored in computer readable form, which led to the development of database management systems (DBMSs)
- The development of the WWW resulted in search engines
- Enabling techniques
    - Tools, to automatically, or semi-automatically extract contents and features contained in multimedia data
    - Multidimensional indexing to handle multimedia feature vectors
    - Similarity metrics, for multimedia retrieval instead of exact match
    - Storage subsystems, redesigned to cope with the requirements of large size and high bandwidth and meet real time requirements
    - Interaction mechanism between user and computer system
    - The user interface, designed to allow flexible queries in different media types and provide multimedia presentation.
- Information retrieval is concerned with developing algorithms and models for retrieving information from document repositories according to an information need
- Information retrieval deals with the representation, storage, organisation of, and access to information items
- Major issues in information
    - Content analysis
    - Indexing
    - Ranking
    - Interaction
    - Results presentation (visualisation)

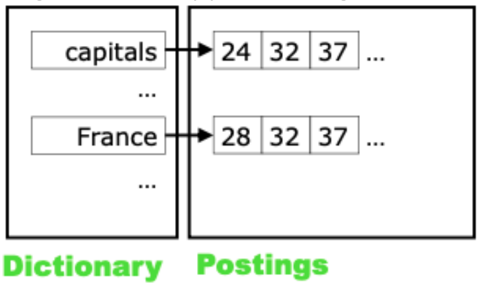


Document representation

- **Terms** – hello, world, information…
- **Boolean vector –** dictionary/vocabulary, the ith dimension indicates whether the ith term in the vocabulary appears in the given document e.g. capital AND France
- **Index (inverted index)**
  - For each term T, we must store a set of all documents that contain T
  - **Term-document incidence matrix**

	Doc 1	**Doc 2**	**Doc 3**
And	0	0	1
Are	0	0	1
Called	1	0	1
Capital	1	1	0
Capitals	0	0	1
France	1	1	1

  - *1 = word appears, 0 = word does not appear*
  - *Query: Capital AND FRANCE → (1, 1, 0)*
  - Search is very easy but not very efficient as resultant matrix has many cells
  - The good news is that this matrix is very sparse (lots of 0's, only a few 1s)
  - Idea: just store the 'hits' (term incidences) using lists
  - **Inverted file**
  - 

  - Main advantage – easy and efficient search
  - Disadvantage – storage (10%-100% of doc size) and modifications
  - Often other information is stored as well to support advanced queries (e.g. position for phrases) or to speed up the search process (e.g. frequency for query optimisation)
  - Bag of words approach – ignores word order, what terms should go into dictionary?
  - 

  - **Dictionary:** usually kept in memory (fast)