Data Mining

Text Mining (Part A)

Dr. Jason T.L. Wang, Professor

Department of Computer Science

New Jersey Institute of Technology

Text Databases and IR

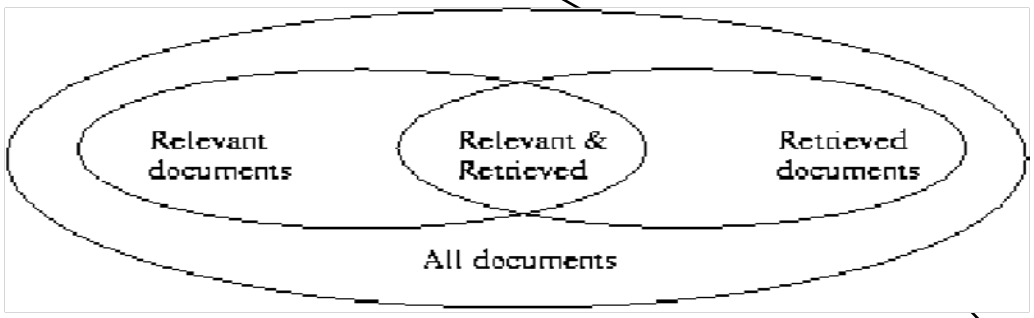
- Text databases (document databases)
  - Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, Web pages, etc.
  - Data stored is usually *semi-structured*
  - Traditional information retrieval (IR) techniques become inadequate for the increasingly vast amounts of text data
- Information retrieval (IR)
  - A field developed in parallel with the database management field
  - Information is organized into (a large number of) documents
  - Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database (DB) management
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects.
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance.

- IR and DB technologies are used together to build modern search engines like Google.

Basic Measures for Information Retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query

$$\text{Precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|}$$

- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$\text{Recall} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|}$$

Keyword-Based Retrieval

- A document is represented by a set of keywords.
- Queries may use expressions of keywords.
  - E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle

  IR systems perform keyword matching.
- Major difficulties of the keyword-based model
  - Synonymy: A keyword *T* (e.g. repair) does not appear anywhere in the document, even though the document is closely related to *T* (e.g. the document is about maintenance)
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining (data mining vs. coal mining)

  Possible solutions: maintain a thesaurus or ontology

Keyword Identification and Extraction in Texts

Question: Given a database of documents, how to identify and extract keywords from the documents?

Answer: There are several ways. We describe one method here.

First, remove words in a stop list from each document.

Stop list

- Set of words that are deemed “irrelevant”, even though they may appear frequently.
- E.g., *a, the, of, for, with*, etc.
- Stop lists may vary when document sets vary.

Second, represent words by word stems.

Word stem

- Several words are small syntactic variants of each other since they share a common word stem.
- E.g., *drug*, *drugs, drugged*
- We represent these words by the word stem.

Keyword Identification and Extraction in Texts

Third, use the tf-idf method to identify keywords in the database **D** of N documents.

Here, the term frequency tf(t, d) is the number of times the term (or word) t occurs in the document d.

The inverse document frequency idf(t) = log (N/df(t)) where df(t) is the document frequency of t, which is the number of documents containing t.

Then tf-idf(t, d) = tf(t, d) × idf(t)

Run the tf-idf algorithm on all words/terms in every document in the database **D**. Rank the words/terms based on their tf-idf scores. Pick the top-k unique words/terms (k is user-determined) with the largest tf-idf scores and use these k words/terms as the keywords for the database **D**.

Vector Space Model for IR

Each document d is represented by a

k-dimensional vector where a dimension corresponds to a keyword/term t and the value in that dimension is tf-idf(t, d); if t does not occur in d, the value in that dimension is 0.

Similarity measure: measure the closeness of two documents (vectors)

- Cosine distance:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$$

End of
Text Mining Module (Part A)