Module 5

Text Mining

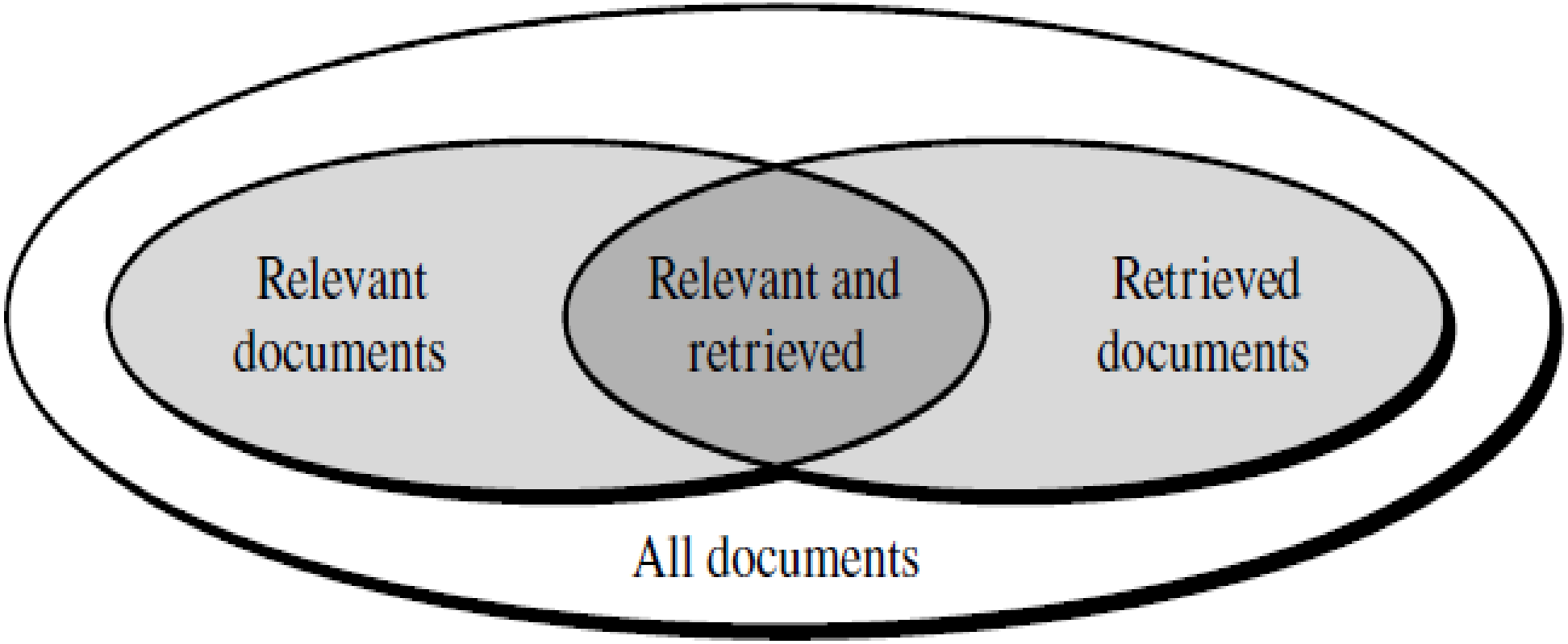
Text Mining

- Text mining has become an increasingly popular and essential theme in data mining.
- Users need tools to compare different documents, rank the importance and relevance of the documents, or find patterns and trends across multiple documents.

Text Data Analysis and Information Retrieval

- Information retrieval (IR) is a field that has been developing in parallel with database systems for many years
- Information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents.
- **Basic Measures for Text Retrieval: Precision and Recall**

Let the set of documents relevant to a query be denoted as {Relevant}, and the set of documents retrieved be denoted as {Retrieved}. The set of documents that are both relevant and retrieved is denoted as {Relevant}{Retrieved}


Relevant documents
Relevant and retrieved
Retrieved documents
All documents

- Precision: This is the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses). It is formally defined as

Recall: This is the percentage of documents that are relevant to the query and were, in fact, retrieved. It is formally defined as

$$F\_score = \frac{recall \times precision}{(recall + precision)/2}.$$

Text Retrieval Methods

- Retrieval methods fall into two categories:
  - Document selection problem
  - Document ranking problem.

Document selection problem

- In document selection methods, the query is regarded as specifying constraints for selecting relevant documents.
- A typical method of this category is the Boolean retrieval model, in which a document is represented by a set of keywords and a user provides a Boolean expression of keywords, such as “car and repair shops,” “tea or coffee,” or “database systems but not Oracle.”

Document ranking problem

- Document ranking methods use the query to rank all documents in the order of relevance.
- For ordinary users and exploratory queries, these methods are more appropriate than document selection methods.
- Most modern information retrieval systems present a ranked list of documents in response to a user’s keyword query.
- There are many different ranking methods

Vector space model.

- Most popular approach for information retrieval.
- The basic idea of the vector space model is the following: We represent a document and a query both as vectors in a high-dimensional space corresponding to all the keywords and use an appropriate similarity measure to compute the similarity between the query vector and the document vector.
- The similarity values can then be used for ranking documents.

- Tokenization: The first step in most retrieval systems is to identify keywords for representing documents, a preprocessing step often called tokenization.
- Stop list: To avoid indexing useless words, a text retrieval system often associates a stop list with a set of documents. A stop list is a set of words that are deemed “irrelevant.” For example, a, the, of, for, with, and so on are stop words.
- Word Stem: A group of different words may share the same word stem, where the words in a group are small syntactic variants of one another and collect only the common word stem per group. For example, the group of words drug, drugged, and drugs, share a common word stem, drug.

“How can we model a document to facilitate information retrieval?”

- Starting with a set of “d” documents and a set of “t” terms, we can model each document as a vector v in the t dimensional space which is why this method is called the vector-space model.
- freq(d, t)- the term frequency be the number of occurrences of term t in the document d
- TF(d, t) - The (weighted) term-frequency matrix measures the association of a term t with respect to the given document d: it is generally defined as 0 if the document does not contain the term, and nonzero otherwise.

- Cornell SMART system uses the following formula to compute the (normalized) term frequency:

$$TF(d,t) = \begin{cases} 0 & \text{if } freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

- Inverse document frequency (IDF), that represents the scaling factor, or the importance, of a term t. If a term t occurs in many documents, its importance will be scaled down due to its reduced discriminative power.

$$IDF(t) = \log \frac{1 + |d|}{|d_t|},$$

where d is the document collection, and d_t is the set of documents containing term t.

- In a complete vector-space model, TF and IDF are combined together, which forms the TF-IDF measure:

$$TF\text{-}IDF(d,t) = TF(d,t) \times IDF(t).$$

“How can we determine if two documents are similar?”

- A representative metric is the cosine measure, defined as follows.
- Let v1 and v2 be two document vectors. Their cosine similarity is d

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

where the inner product $v_1 \cdot v_2$ is the standard vector dot product, defined as $\Sigma_{i=1}^{t} v_{1i} v_{2i}$, and the norm $|v_1|$ in the denominator is defined as $|v_1| = \sqrt{v_1 \cdot v_1}$.

Text Indexing Techniques

- There are several popular text retrieval indexing techniques, including inverted indices and signature files.
- An **inverted index** is an index structure that maintains two hash indexed or B+-tree indexed tables: document_table and term_table,

- Document_table consists of a set of document records, each containing two fields: doc_id and posting_list, where posting list is a list of terms (or pointers to terms) that occur in the document, sorted according to some relevance measure.
- Term_table consists of a set of term records, each containing two fields: term id and posting list, where posting list specifies a list of document identifiers in which the term appears.

- A **signature file** is a file that stores a signature record for each document in the database.
- Each signature has a fixed size of b bits representing terms.
- A simple encoding scheme goes as follows.
- Each bit of a document signature is initialized to 0.
- A bit is set to 1 if the term it represents appears in the document.
- A signature S1 matches another signature S2 if each bit that is set in signature S2 is also set in S1.

Query Processing Techniques

Once an inverted index is created for a document collection, a retrieval system can answer a keyword query quickly by looking up which documents contain the query keywords.

- When examples of relevant documents are available, the system can learn from such examples to improve retrieval performance.
- This is called relevance feedback and has proven to be effective in improving retrieval performance.

- When we do not have such relevant examples, a system can assume the top few retrieved documents in some initial retrieval results to be relevant and extract more related keywords to expand a query.
- Such feedback is called pseudo-feedback or blind feedback.
- One major limitation of many existing retrieval methods is that they are based on exact keyword matching.
- Due to the complexity of natural languages, keyword based retrieval can encounter two major difficulties.

- The first is the synonymy problem: two words with identical or similar meanings may have very different surface forms.
  - For example, a user’s query may use the word “automobile,” but a relevant document may use “vehicle” instead of “automobile.”
- The second is the polysemy problem: the same keyword, such as mining, or Java, may mean different things in different contexts.