Pre-procesing techniques for text classification

An introduction

Introduction

- Text mining refers to data mining using text documents as data.
- Most text mining tasks use Information Retrieval (IR) methods to pre-process text documents.
- These methods are quite different from traditional data pre-processing methods used for relational tables.
- Web search also has its root in IR.

Information Retrieval (IR)

- Conceptually, IR is the study of finding needed information. I.e., IR helps users find information that matches their information needs.
  - Expressed as queries
- Historically, IR is about document retrieval, emphasizing document as the basic unit.
  - Finding documents relevant to user queries
- Technically, IR studies the acquisition, organization, storage, retrieval, and distribution of information.

Boolean model

- Each document or query is treated as a **"bag" of words** or **terms**. Word sequence is not considered.
- Given a collection of documents D, let $V = \{t_1, t_2, ..., t_{|V|}\}$ be the set of distinctive words/terms in the collection. V is called the vocabulary.
- A weight $w_{ij} > 0$ is associated with each term t_i of a document $\mathbf{d}_j \in D$. For a term that does not appear in document $\mathbf{d}_j$, $w_{ij} = 0$.

$$\mathbf{d}_j = (w_{1j}, w_{2j}, ..., w_{|V|j}),$$

Vector space model

- Documents are also treated as a “bag” of words or terms.
- Each document is represented as a vector.
- However, the term weights are no longer 0 or 1. Each term weight is computed based on its frequency in the documents

Weighting schema: TF-IDF

- Documents are also treated as a "bag" of words or terms.
- Each document is represented as a vector
- The term weight is computed in a sophisticated way
- Term Frequency (TF) Scheme: The weight of a term t_i in document $\mathbf{d}_j$ is the number of times that t_i appears in $\mathbf{d}_j$, denoted by f_{ij}. Normalization may also be applied.

TF-IDF term weighting scheme

(normalization with max)

- The most well known weighting scheme
  - TF: still **term frequency**
  - IDF: **inverse document frequency**.

  N: total number of docs

  df_i: the number of docs that t_i appears.
- The final TF-IDF term weight is:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

$$idf_i = \log \frac{N}{df_i}$$

$$w_{ij} = tf_{ij} \times idf_i.$$

TF-IDF term weighting scheme

(normalization with sum)

- The most well known weighting scheme
  - TF: still **term frequency**
  - IDF: **inverse document frequency**.

  N: total number of docs

  df_i: the number of docs that t_i appears.

$$tf_{ij} = \frac{f_{ij}}{\sum_{i=1}^{|V|} f_{ij}}$$

$$idf_i = \log \frac{N}{df_i}$$

- The final TF-IDF term weight is:

$$w_{ij} = tf_{ij} \times idf_i.$$

Example dataset

- d1 = {machine, learning, support vector, machine, machine, data, tree}
- d2 = {data, mining, associat, classifier, classifier, data, data, associat}
- d3 = {mining, decision, tree, decision}
- d4 = {associat, mining, data, mining,}
- d5 = {decision, tree, classifier}

Boolean representation

	associat	classifier	data	decision	learning	machine	mining	support	tree	vector
d1	0	0	1	0	1	1	0	1	1	1
d2	1	1	1	0	0	0	1	0	0	0
d3	0	0	0	1	0	0	1	0	1	0
d4	1	0	1	0	0	0	1	0	0	0
d5	0	1	0	1	0	0	0	0	1	0

Vector space representation

	associat	classifier	data	decision	learning	machine	mining	support	tree	vector
d1	0	0	1	0	1	3	0	1	1	1
d2	2	2	3	0	0	0	1	0	0	0
d3	0	0	0	1	0	0	1	0	1	0
d4	1	0	1	0	0	0	2	0	0	0
d5	0	1	0	1	0	0	0	0	1	0

TF-IDF representation

(normalization with max)

	associat	classifier	data	decisio n	learni ng	machine	mining	suppor t	tree	vector
d1	0	0	0.17	0	0.54	1.61	0	0.54	0.17	0.54
d2	0.61	0.61	0.51	0	0	0	0.17	0	0	0
d3	0	0	0	0.92	0	0	0.51	0	0.51	0
d4	0.46	0	0.46	0	0	0	0.51	0	0	0
d5	0	0.92	0	0.92	0	0	0	0	0.51	0

Text pre-processing

- Word (term) extraction: easy
- Erase infrequent words
- Stopwords removal
- Stemming
- Frequency counts and computing TF-IDF term weights.

Stopwords removal

- Many of the most frequently used words in English are useless in IR and text mining – these words are called *stop words*.
  - the, of, and, to, ….
  - Typically about 400 to 500 such words
  - For an application, an additional domain specific stopwords list may be constructed
- Why do we need to remove stopwords?
  - Reduce indexing (or data) file size
    - stopwords accounts 20-30% of total word counts.
  - Improve efficiency and effectiveness
    - stopwords are not useful for searching or text mining
    - they may also confuse the retrieval system.

Erase infrequent words

- Erase all the words that are infrequents
  - For instance erase words with a frequency lesser or equal to 2
- Avoid very big feature space

Stemming

- Techniques used to find out the root/stem of a word. E.g.,
  - user
  - users
  - used
  - using

  engineering
  engineered
  engineer
- stem: use engineer

Usefulness:

- improving effectiveness of IR and text mining
  - matching similar words
  - Mainly improve recall
- reducing indexing size
  - combing words with same roots may reduce indexing size as much as 40-50%.

Basic stemming methods

Using a set of rules. E.g., (Porter Stemming Method)

- remove ending
  - if a word ends with a consonant other than s, followed by an s, then delete s.
  - if a word ends in es, drop the s.
  - if a word ends in ing, delete the ing unless the remaining word consists only of one letter or of th.
  - If a word ends with ed, preceded by a consonant, delete the ed unless this leaves only a single letter.
  - ……
- transform words
  - if a word ends with “ies” but not “eies” or “aies” then “ies --> y.”

Summary

- We only give a VERY brief introduction to IR techniques to represent and pre-processing the data
- IR is a very interesting field in which many techniques are developed to store, manage and analyze information
- Many other interesting topics are not covered, e.g.,
  - Web search
    - Index compression
    - Ranking: combining contents and hyperlinks
  - Web page pre-processing
  - Combining multiple rankings and meta search
  - Web spamming