Data preparation for document data

Document representation

- A document might be modeled in different ways
  - The choice heavily affects the quality of the mining result
- The most common representation models a document as a set of features
  - Each feature might represent a set of characters, a word, a term, a concept

Document processing

- It is the activity to generate a structured data representation of document data
- It includes five sequential steps
  - Document splitting
  - Tokenization
  - Case normalization
  - Stopword removal
  - Stemming

Document splitting

- Based on the data analytics goal, documents can be split into
  - sentences, paragraphs, or analyzed in their entire content
- Short documents are typically not split
  - e.g., emails or social posts
- Long documents can be
  - broken up into sections or paragraphs
  - analyzed as a whole

Tokenization

- It is the process of breaking text into sentences or text into tokens (i.e., words)
  - Identify sentence boundaries based on punctuation, capitalization
  - Separate words in sentences
  - Language-dependent

Case normalization

- This step converts each token to completely upper-case or lower-case characters
  - Capitalization helps human readers differentiate, for example, between nouns and proper nouns and can be useful for automated algorithms as well
  - However, an upper-case word at the beginning of the sentence should be treated no differently than the same word in lower case appearing elsewhere in a document

Stemming

- Reduce a word to its root form (i.e., the **stem**)
  - It includes the identification and removal of prefixes, suffixes, and pluralization
- It operates on a single word without knowledge of the context
  - It cannot discriminate between words which have different meanings depending on the part of speech
- Stemmers are
  - Easy to implement
  - Available for most spoken languages
  - Run significantly faster than lemmatization and POS tagging algorithms

Stopword elimination

- "Stop words" refers to the most common words in a language
  - E.g., prepositions, articles, conjunctions in English
- Stop words are filtered out before or after processing of textual data
  - They are likely to have little semantic meaning

Stopword elimination

- There is no single universal list of stop words used by all natural language processing tools
- Any group of words can be chosen as the stop words for a given purpose
  - different search engines use different stop word lists
  - Some of them remove lexical words, such as "want", from a query in order to improve performance
- Some tools specifically avoid removing these stop words to support phrase search

Weighted document representation

Text representation: feature vectors

- Most data mining algorithms are unable to directly process textual data in their original form
  - documents are transformed into a more manageable representation
- Documents are represented by feature vectors
- A feature is simply an entity without internal structure
  - A dimension of the feature space
- A document is represented as a vector in this space
  - a collection of features and their weights

Example

- Each document becomes a term vector
  - each term is a component (attribute) of the vector
  - the value of each component is the number of times the corresponding term occurs in the document

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

From: Tan,Steinbach, Kumar, Introduction to Data Mining, McGraw Hill 2006

Bag-of-word representation

- All words in a document are considered as separate features
  - the dimension of the feature space is equal to the number of different words in the entire document collection
- The feature vector of a document consists of a set of weights, one for each distinct word
- The methods for giving weights to the features may vary

Weighting schemes

- Binary
  - One, if the corresponding word is present in the document
  - Zero, otherwise
  - Occurrences of all words have the same importance
- Simple document frequency
  - The number of times in which the corresponding word occurs in the document
  - Most frequent words are not always representative of the document content

Weighting schemes

- More complex weighting schemes are possible to take into account the frequency of the word
  - in the document
  - in the section/paragraph
  - in the category (for indexed documents)
  - in the collection of documents

Weighting schemes

- Term frequency inverse document frequency (tf-idf)
  - Tf-idf of term t in document d of collection D (consisting of m documents)

    $$\text{tf-idf}(t) = \text{freq}(t, d) * \log(m/\text{freq}(t, D))$$

  - Terms occurring frequently in a single document but rarely in the whole collection are preferred
- Suitable for
  - A single document consisting of many sections or subsections
  - A collection of *heterogeneous* documents

Tf-idf matrix example

- Most common words (e.g., “model”) have low values
- Peculiar words (e.g., “medlin”, “micro”, “methodolog”) have high values

major	malform	materi	matric	matrix	mean	measur	mechan	medicin	medium	medlin	method	methodolog	micro	microarch...	migrat	mo	model	molecular	morbid	moreov	mortal
0	0	0.153	0.051	0.021	0	0	0	0	0	0	0.051	0.069	0.072	0	0.020	0	0.034	0.072	0	0.072	0.063
0.032	0.032	0.048	0.032	0.020	0.032	0.032	0.032	0.064	0.032	0.032	0.048	0.043	0.023	0.032	0.018	0.032	0.022	0.023	0.095	0.023	0.033
0	0	0	0	0.016	0	0.077	0.077	0	0	0	0.039	0.026	0	0.077	0.007	0.077	0	0	0	0	0.016
0.085	0.171	0	0	0	0	0	0	0	0.171	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.153	0.051	0.021	0	0	0	0	0	0	0.051	0.069	0.072	0	0.020	0	0.034	0.072	0	0.072	0.063
0	0	0	0.052	0	0.105	0	0	0.052	0	0.052	0	0.035	0	0	0.020	0	0.035	0	0	0	0.022
0.093	0	0	0	0.039	0	0	0	0.093	0	0.093	0	0	0	0	0.018	0	0	0	0	0	0
0.077	0	0.154	0	0.032	0	0	0	0.077	0	0.077	0	0	0	0	0.030	0	0.052	0	0	0	0.032

Weighting schemes

- Document-Term matrix X
  - Local weight l_{ij}
  - Global weight g_j

- $X_{ij} = l_{ij} * g_j$