Data Mining II / Adv. Topics in Data Science

4 Hands On: Information Retrieval and Text Mining

Load the packages tm, text2vec.

4.1 Information Retrieval

1. Using the functions VectorSource() and Corpus(), start by creating a corpus with three "documents" containing the following text:

- *"Mining is important for finding gold"*
- *"Classification and regression are data mining"*
- *"Data mining deals with data"*

Then, use the function DocumentTermMatrix() to represent the documents.

(a) Use the functions nDocs(), nTerms(), Terms() to get some infomation on the DocumentTermMatrix you have created.

(b) If you inspect the DocumentTermMatrix, what information does it give you? What is the document representation model employed by default? If you want to get the complete DocumentTermMatrix, you should use the function as.matrix

(c) Use the function weightBin() on the original document term matrix to represent the documents with a vector space model, but with a binary scheme.

(d) Use the function weightTfIdf() on the original document term matrix to represent the documents with a vector space model, but with TF-IDF scheme.

(e) Did any of the terms get a zero value for all documents? Which ones? What does this tell you about the discriminative power of the term?

(f) Analyze the cosine similarity between the three documents, in each weighting scheme. You can use the function sim2 from the package text2vec on each matrix.

2. Rank the above documents given the query "data mining" by the cosine similarity of the query to each document:

(a) using binary scheme

(b) using TF scheme

(c) using TF-IDF scheme

4.2 Text Mining

Pre-processing steps

3. Let us now use a set of documents which represent news from Reuters news agency, related with crude oil. These documents are available on the tm package and are stored as XML files following the format used by Reuters.

(a) Load the above referred files by executing the following code:

```
## The place (subdirectories within the tm package) where the files are
stored reut21578 <- system.file("texts", "crude", package =  "tm")
## Now creating a corpus using the files in a certain
directory reuters <- VCorpus(DirSource(reut21578),
                   readerControl = list(reader = readReut21578XMLasPlain))
```

Note: For this particular set of documents, tm package has already the corpus created, and you can load it by executing data(crude)

(b) Inspect the first text of the loaded corpus.

(c) Load the package wordcloud to obtain a graphical representation of the terms in the corpus.

(d) Use the function tm_map to apply the following transformations to the texts forming a corpus:

- strip white space
- convert everything to lowercase
- remove english stopwords
- obtain words stem (keeping only the “root” of each word)
- remove punctuation, by taking into account that intra-words contractions and intra-words dashes should be preserved.

(e) Obtain a graphical representation of the frequencies of terms in the transformed corpus. Is it too different from the original representation?

(f) Convert the transformed corpus into a Document Term Matrix and inspect a few entries of the matrix.

(g) Use the function FindFreqTerms for obtaining the terms that occur more than 10 times.

(h) Use the function findAssocs for obtaining the terms with a correlation higher than 0.8 with the term “opec”, which stands for “Organization of the Petroleum Exporting Countries”.