Cairo University
Faculty of Engineering
Computer Engineering Department
CMPN451 - Big Data Mining

Lab 8 - Text Analysis

Introduction:
Text analysis is the process of examining large collections of documents with an objective to extract new information, discover answers for specific research questions, identify facts, relationships, assertions that would otherwise remain buried in the mass of textual big data. One extracted, this information is converted into a structural representation that can be further analyzed and processed. It is estimated that as much as 80% of the world's data is unstructured, at the same time when we find that most work in data analysis is devoted to structured data. In this lab, we will explore the potential of R packages to analyze unstructured text using text mining "tm" R package.

Lab Requirements:

1. Review the lecture of Text Analysis. The slides are uploaded on the drive and the video of the lecture is uploaded on the course's playlist on YouTube.

Lab Instructions:

1. This is a coding **individual** lab.
2. You will find the documentation of the 'tm' library. It will be helpful for this lab.

Lab Deliverables:

1. You will submit R-script containing the solution of the questions in the lab document.

Main points in Text Mining:

(1) **Importing data:**

a) The main structure for managing documents in 'tm' is a so-called **Corpus**, representing a collection of text documents.

b) Every operation using any 'tm' functionality applied only on a Corpus object.

c) Example: `pcorpus <- Corpus(VectorSource(dataframe))`

(2) **Data Preprocessing:**

a) The text mining library 'tm' supports many transformation functions on a Corpus object.

b) For example `pcorpus <- tm_map(pcorpus, tolower)` transforms each upper case character in the corpus to lower case.

c) Find other available transformation functions using help widget

```
?tm_map
 tm::getTransformations
```

d) You can also open the tm.pdf document attached with the lab under section *Transformation*.

(3) **Term-Document Matrix:**

a) A *document-term matrix* or *term-document matrix* is a mathematical matrix that describes the frequency of terms that occur in a collection of documents.

b) In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms.

c) To create a document-term matrix, we can use this line:

```
pdtm <- DocumentTermMatrix(pcorpus)
```

d) In text mining, when you use a bag of words (BOW) approach, ignoring terms that have a document frequency lower than a given threshold (we call them sparse terms), can help generalization and prevent over-fitting.

```
str(inspect(removeSparseTerms(pdtm, 0.999))) // to show the matrix
pdtm <- removeSparseTerms(pc, 0.999)
```

e) To get terms of high frequency you can choose this function

```
pfreq <- findFreqTerms(pdtm, 70)
```

f) You can also find associated terms using this function

```
findAssocs(pdtm, "feel", 0.1)
```

g) Note: All these operations are done on the term document matrix, to be able to apply any normal functionality on the data like (sort, head, colSums) you must firstly convert it to normal matrix like this

```
pdtm2 <- as.matrix(pdtm)
```

```
pfrequency <- colSums(pdtm2)
head(pfrequency)
```

(4) **World Cloud:**

a) Using the library “wordcloud” displays a graphic representation of the words frequency extracted from the document term matrix.

b) Every word has a different size, bigger size words means they have higher frequencies.

c) This can be done through these functions

```
pwords <- names(pfrequency)
wordcloud(pwords, pfrequency)
```

Lab Questions:

You are required to apply what you learned in this lab “Text Mining Techniques” on the *Movies* dataset.

1. Import the data in ‘movie_reviews.csv’ into a dataframe.
2. Inspect the dataframe to get familiar with the dataset.
3. Construct a corpus object from the ‘text’ column in the dataframe.
4. Convert all the words of the corpus from uppercase to lowercase
5. Remove punctuation marks, numbers, and white spaces from the corpus.
6. Remove stopping words (in English language) from the corpus.
7. Construct the document term matrix.
8. Inspect the document term matrix and determine its sparsity percentage.
9. Remove sparse terms below a sparsity threshold 0.999
10. Find words of frequencies higher than 65.
11. Find the associations of words “*movie” and “live”.*
12. Construct a normal matrix from the term-document matrix and **sort** the words by their frequency in a descending order.
13. Display the most frequent five words in the corpus.
14. Display the word cloud of **the first 100 words**.