Practical No. 4

Approach Used & Reason for using the Approach:

I have used the approach to select entities through the use of openNLP package available in R as compared to Stanford core NLP which is much slower and more difficult to install. Also, I executed the code on my Laptop but it can be executed on Spark using R with more data and faster results. It involves the following steps:

Problem faced: I needed to install openNLP version 0.9 instead of 0.15 presently available because some methods didn't execute on it.

1. Installing the different packages in R Studio.
2. Creating a text string from all the different columns.
3. Creatng a word, sentence and parts of Speech annotator.
4. Create annotated plain text documents from plain text and collections of annotations for this text.
5. Create Person, Location, Date and Organization annotations.
6. Create a pipeline from all anotations.
7. Create a method to extract entities from an AnnotatedPlainTextDocument
8. Use appropriate functions to extract entities of the correct type.
9. Create a data frame from the tabulated contents of the Entites output.
10. Plot the output using Google Visualization.

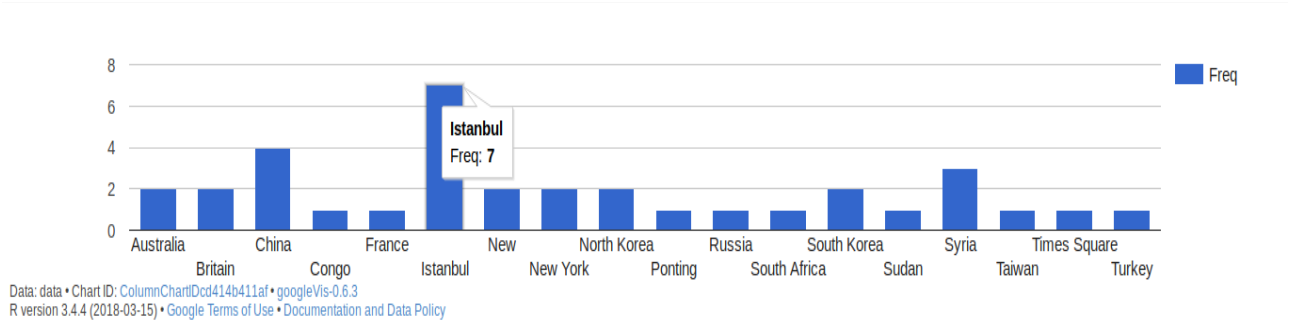
Code:

```
a <- rediff_realtime_news_201701_201703[,4, drop=FALSE]

# Script to scrap Reuters
reutersreports <- rediff_realtime_news_201701_201703[grep("Reuters",
rediff_realtime_news_201701_201703$source), ]
reuterscities <- reutersreports
reuterscities$location <- gsub("(\\b(([a-z][a-z]+\\s[a-z]+))|(([A-Z][a-z]+))|(([a-z]+))|(([a-z][A-Z]+))|
(([0-9]+))|(\\W)+|(REUTERS)\\b)|(\\b([A-Z])\\b)", " ", reuterscities$summary)
text <- paste(head(reuterscities$title, 50), sep=" ")

library(rJava) #require(rJava) would do too.
library(openNLP)
library(NLP)
library(magrittr)
library(openNLPmodels.en)
library("googleVis", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")

text <- paste(text, collapse = " ")
print (text)
text <- as.String(text)

## Chunking needs word token annotations with POS tags.
word_ann <- Maxent_Word_Token_Annotator()
sent_ann <- Maxent_Sent_Token_Annotator()
pos_ann <- Maxent_POS_Tag_Annotator()
```

```
pos_annotations <- annotate(text, list(sent_ann, word_ann, pos_ann))
text_annotations <- annotate(text, list(sent_ann, word_ann))
head(text_annotations)


text_doc <- AnnotatedPlainTextDocument(text, text_annotations)
words(text_doc) %>% head(10)

#install.packages("openNLPmodels.en", dependencies=TRUE, repos = "http://datacube.wu.ac.at/")

person_ann <- Maxent_Entity_Annotator(kind = "person")
location_ann <- Maxent_Entity_Annotator(kind = "location")
organization_ann <- Maxent_Entity_Annotator(kind = "organization")
date_ann <- Maxent_Entity_Annotator(kind = "date")
#topic_ann <- Maxent_Entity_Annotator(kind = "money")

pipeline <- list(sent_ann,
                 word_ann,
                 person_ann,
                 location_ann,
                 organization_ann, date_ann)

text_annotations <- annotate(text, pipeline)
text_doc <- AnnotatedPlainTextDocument(text, text_annotations)


# Extract entities from an AnnotatedPlainTextDocument
entities <- function(doc, kind) {
  s <- doc$content
  a <- annotations(doc)[[1]]
  if(hasArg(kind)) {
    k <- sapply(a$features, `[[`, "kind")
    s[a[k == kind]]
  } else {
    s[a[a$type == "entity"]]
  }
}

# Use the function and extract the entity kind
entities(text_doc, kind = "person")
entities(text_doc, kind = "location")
entities(text_doc, kind = "organization")
entities(text_doc, kind = "date")

dfp <- data.frame(table(entities(text_doc, kind = "person")))
Barp <- gvisColumnChart(dfp)
plot(Barp)

dfl <- data.frame(table(entities(text_doc, kind = "location")))
```

```
Barl <- gvisColumnChart(dfl)
plot(Barl)

dfo <- data.frame(table(entities(text_doc, kind = "organization")))
Baro <- gvisColumnChart(dfo)
plot(Baro)

dfd <- data.frame(table(entities(text_doc, kind = "date")))
Bard <- gvisColumnChart(dfd)
plot(Bard)
```

SAMPLE OUTPUT:



Conclusion:

Although the problem faced wasn't very difficult it could proved to be if, had I not seeked professional help on stackoverflow.com considering the lack of information available on the R official website and openNLP website regarding the methods and their new versions and deprecations.

However, this code could be improved much if this code could be executed on multicore environment like that of Google Collab using the famed Python Multicore libraries.

Nonetheless it was good to have a basic understanding of R and how it varies and differs with Python.