Task 2

1. Describe the Reuters-21578 corpus

Reuters-21578 is a test collection for text classification research that is a multi-class, multi-label dataset. This dataset contains 90 classes, 7769 training files, and 3019 test files is a ModApte subdirectory of the Reuters-21578 benchmark. The Reuters-21578 dataset was originally collected and tagged by the Carnegie Group and Reuters in 1987 during the development of the CONSTRUE text classification system, and later by AT&T Labs Research in September 1997. released in February, with David D. Lewis as the lead publisher.

2. Describe how each document is represented in your implementation.

```
> data(Reuters21578)
> class(Reuters21578)
[1] "VCorpus" "Corpus"
> head(Reuters21578)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 6
> summary(Reuters21578)
      Length Class             Mode
1     2      PlainTextDocument list
2     2      PlainTextDocument list
3     2      PlainTextDocument list
4     2      PlainTextDocument list
5     2      PlainTextDocument list
...
```

We import the Reuters-21578 as Vcorpus, which contains Metadata and Content. The metadata attribute contains author, datetime stamp, description, heading id, language, origin, lewissplit, cgisplit, oldid, topics_cat, places, people, orgs, exchanges. The content is the raw data.

The data structure in the `tm` package that mainly manages documents is called Corpus, which represents a collection of documents. The corpus is divided into a dynamic corpus (Volatile Corpus) and a static corpus (Permanent Corpus). A dynamic corpus will be stored in memory as an R object and can be generated by either VCorpus() or Corpus(). The dynamic corpus, on the other hand, is stored as an R external file and can be generated using the PCorpus() function.

3. Describe the whole procedure on applying LDA to this corpus to perform topic modeling.

1. Import the dataset

2. Pre-processing the dataset, including transforming the content to lower case, striping whitespace, removing stopwords, punctuation and numbers, and steming document.
3. Calculate the BOW/TF-IDF document term matrix, `bowdtm` and `tfidfdtm`
4. Reducint the dimension with `tfidfdtm`
5. Calculate the word cloud with `bowdtm`
6. Apply LDA analysis.

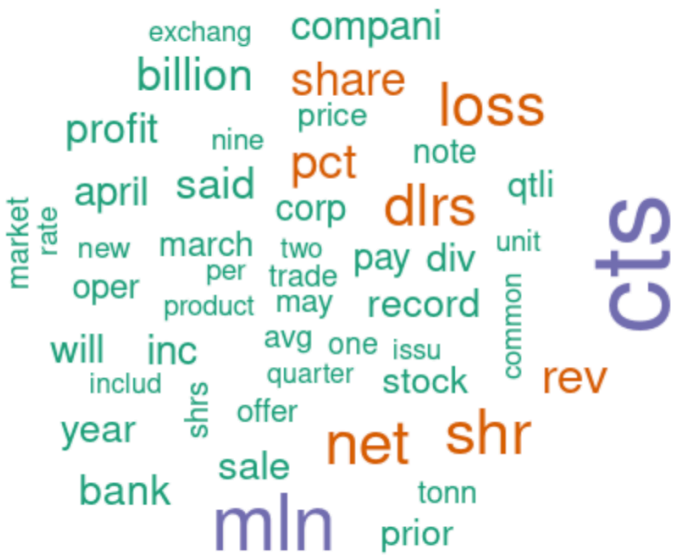
4. Describe the parameter setting that you use in the LDA and explain their meanings.

```
result <- LDA(bowdtm, k, method="Gibbs", control=list(iter = 25, verbose = 25, alpha = 0.1))
```

`bowdtm`: The document term matrix with BOW method.
`k`: number of topics
`method=Gibbs`: Applying Gibbs sampling
`control=list(iter = 25, verbose = 25, alpha = 0.1)`: inference via 25 iterations.

5. Describe the output of your code and visualize the obtained topics in appropriate ways.

Draw the word cloud from the TF-IDF document term matrix.



We can see that "cts", "mln" is the most frequent words.

```
> bowdtm <- bowdtm[slam::row_sums(bowdtm) > 0, ]
> k <- 20
> result <- LDA(bowdtm, k, method="Gibbs", control=list(iter = 25, verbose = 25, alpha = 0.1
K = 20; V = 32697; M = 19042
Sampling 25 iterations!
Iteration 25 ...
Gibbs sampling completed!
> result
A LDA_Gibbs topic model with 20 topics.
> terms(result, 10)
      Topic 1   Topic 2   Topic 3   Topic 4   Topic 5   Topic 6   Topic 7   Topic 8      To
 [1,] "said"    "said"    "said"    "said"    "dlrs"    "billion" "said"    "tonn"       "b
 [2,] "govern"  "will"    "trade"   "trade"   "mln"     "bank"    "market"  "said"       "s
 [3,] "econom"  "compani" "japan"   "reuter"  "said"    "pct"     "rate"    "mln"        "c
 [4,] "japan"   "new"     "offici"  "futur"   "year"    "franc"   "dollar"  "wheat"      "l
 [5,] "offici"  "reuter"  "state"   "price"   "quarter" "said"    "bank"    "export"     "b
 [6,] "minist"  "car"     "import"  "pct"     "compani" "year"    "trade"   "reuter"      "c
 [7,] "year"    "trade"   "japanes" "new"     "sale"    "mln"     "analyst" "agricultur" "w
```

```
 [8,] "will"    "motor"   "unit"    "cent"     "earn"    "reuter"  "currenc" "year"       "i
 [9,] "west"    "exchang" "will"    "tonn"     "share"   "foreign" "dealer"  "grain"      "c
[10,] "japanes" "market"  "reuter"  "contract" "report"  "mark"    "exchang" "crop"       "r
      Topic 16  Topic 17  Topic 18  Topic 19 Topic 20
 [1,] "said"    "said"    "said"    "mln"    "pct"
 [2,] "tax"     "compani" "compani" "cts"    "year"
 [3,] "billion" "dlrs"    "will"    "net"    "said"
 [4,] "budget"  "reuter"  "reuter"  "loss"   "billion"
 [5,] "bill"    "share"   "inc"     "dlrs"   "februari"
 [6,] "stg"     "mln"     "corp"    "shr"    "januari"
 [7,] "hous"    "corp"    "system"  "reuter" "rose"
 [8,] "dlrs"    "inc"     "new"     "profit" "rise"
 [9,] "reuter"  "group"   "servic"  "rev"    "last"
[10,] "bank"    "will"    "comput"  "oper"   "month"
```

We should remove empty rows in `bowdtm` and set number of topics to 20. Then we can compute the LDA model, inferencing via 25 iterations of Gibbs sampling.

We can see the 10 most likely terms within the term probabilities beta of the inferred topics.

We took eight sample documents:

```
examples <- c(2, 100, 200, 400, 800, 1000, 1200, 1400)
lapply(pre_process_reuters[examples], as.character)

theta <- tmposterior$topics
N <- length(examples)

tpExamples <- theta[examples,]
colnames(tpExamples) <- nameOfTopics
vizDataFrame <- melt(cbind(data.frame(tpExamples), document = factor(1:N)), variable.name =
vizDataFrame
```

and get topic proportions form example documents.

6. Attach your code to the ZIP file.

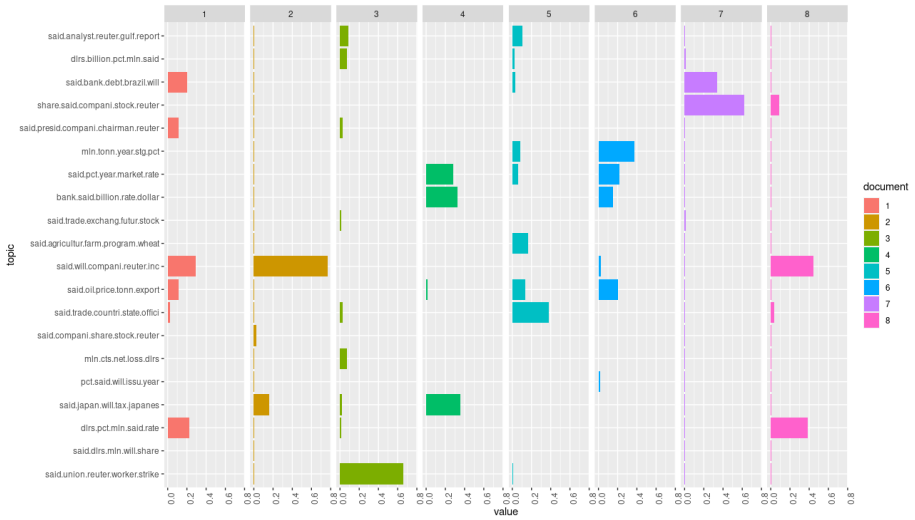

1
2
3
4
5
6
7
8
said.analyst.reuter.gulf.report
dlrs.billion.pct.mln.said
said.bank.debt.brazil.will
share.said.compani.stock.reuter
said.presid.compani.chairman.reuter
mln.tonn.year.stg.pct
said.pct.year.market.rate
bank.said.billion.rate.dollar
said.trade.exchang.futur.stock
said.agricultur.farm.program.wheat
said.will.compani.reuter.inc
said.oil.price.tonn.export
said.trade.countri.state.offici
said.compani.share.stock.reuter
mln.cts.net.loss.dlrs
pct.said.will.issu.year
said.japan.will.tax.japanes
dlrs.pct.mln.said.rate
said.dlrs.mln.will.share
said.union.reuter.worker.strike
topic
0.0
0.2
0.4
0.6
0.8
value
document
1
2
3
4
5
6
7
8

Figure 1: