# CSCI6444 Project3 Report

**First of all, we create a corpus which only has one document in--'TWENTY THOUSAND LEAGUES UNDER THE SEA'.**

```
> FIC<-VCorpus(DirSource(".", ignore.case = TRUE, mode = "text"))
> FIC
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 1
> inspect(FIC)
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 1

[[1]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 577411
```

```
> str(FIC)
List of 1
 $ a.txt:List of 2
  ..$ content: chr [1:12130] "" "TWENTY THOUSAND LEAGUES UNDER THE SEA" "" "by" ...
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2020-04-30 13:31:34"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "a.txt"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 - attr(*, "class")= chr [1:2] "VCorpus" "Corpus"
```

**Then, to get the longest 10 sentences and 10 words in this fiction, we extract the document.**

```
> test1<-FIC[[1]]
> test1
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 577411
> str(test1)
List of 2
 $ content: chr [1:12130] "" "TWENTY THOUSAND LEAGUES UNDER THE SEA" "" "by" ...
 $ meta   :List of 7
  ..$ author       : chr(0)
  ..$ datetimestamp: POSIXlt[1:1], format: "2020-04-30 13:31:34"
  ..$ description  : chr(0)
  ..$ heading      : chr(0)
  ..$ id           : chr "a.txt"
  ..$ language     : chr "en"
  ..$ origin       : chr(0)
  ..- attr(*, "class")= chr "TextDocumentMeta"
 - attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
```

**Before getting the longest sentences and words, we tried some functions first.**

```
> FICdtm<-DocumentTermMatrix(FIC)
> FICdtm
<<DocumentTermMatrix (documents: 1, terms: 14907)>>
Non-/sparse entries: 14907/0
Sparsity           : 0%
Maximal term length: 26
Weighting          : term frequency (tf)
> inspect(FICdtm)
<<DocumentTermMatrix (documents: 1, terms: 14907)>>
Non-/sparse entries: 14907/0
Sparsity           : 0%
Maximal term length: 26
Weighting          : term frequency (tf)
Sample             :
       Terms
Docs     and for had not that  the this  was which with
  a.txt 2366 559 620 881  926 8355  709 1307   730  853
```

```
> str(FICdtm)
List of 6
 $ i       : int [1:14907] 1 1 1 1 1 1 1 1 1 1 ...
 $ j       : int [1:14907] 1 2 3 4 5 6 7 8 9 10 ...
 $ v       : num [1:14907] 1 1 1 1 1 1 1 1 1 1 ...
 $ nrow    : int 1
 $ ncol    : int 14907
 $ dimnames:List of 2
  ..$ Docs : chr "a.txt"
  ..$ Terms: chr [1:14907] "'artocarpus'" "'bread-fruit'" "'seafrog,'" "'these" ...
 - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
> FICtdm<-TermDocumentMatrix(FIC)
> FICtdm
<<TermDocumentMatrix (terms: 14907, documents: 1)>>
Non-/sparse entries: 14907/0
Sparsity           : 0%
Maximal term length: 26
Weighting          : term frequency (tf)
> inspect(FICtdm)
<<TermDocumentMatrix (terms: 14907, documents: 1)>>
Non-/sparse entries: 14907/0
Sparsity           : 0%
Maximal term length: 26
Weighting          : term frequency (tf)
Sample             :
       Docs
Terms   a.txt
  and    2366
  for     559
  had     620
  not     881
  that    926
  the    8355
  this    709
  was    1307
  which   730
  with    853
```

**Now, we process the document.**

(this is part showing how to get the 10 longest words and sentences)

**The 10 longest words:**

```
> countwords<-sapply(FICtdm$dimnames$Terms, nchar)
> dt<-data.frame(word = unlist(FICtdm$dimnames$Terms), length = unlist(countwords))
> dt<-dt[order(dt$length),]
> tail(dt, 10)
                                          word length
venerated--country,             venerated--country,     19
compagnie-nationale,           compagnie-nationale,     20
emperor-holocanthus,           emperor-holocanthus,     20
waters--tuberculated           waters--tuberculated     20
communication--rather         communication--rather     21
harpooner--commander,         harpooner--commander,     21
mohammed-ben-abdallah,       mohammed-ben-abdallah,     22
observed--turritellas,       observed--turritellas,     22
self-confidence--because   self-confidence--because     24
"yes--certainly--perhaps," "yes--certainly--perhaps,"     26
```

**Seems that there are some dashes taken into account. Therefore, we did some cleansing:**

```
> FIClow<-tm_map(FIC, content_transformer(tolower))
> removeNumPunct<-function(x) gsub("[^[:alpha:][:space:]*]", "", x)
> FICcl<-tm_map(FIClow, content_transformer(removeNumPunct))
> FICcldtm<-DocumentTermMatrix(FICcl)
> FICcldtm
<<DocumentTermMatrix (documents: 1, terms: 8827)>>
Non-/sparse entries: 8827/0
Sparsity           : 0%
Maximal term length: 21
Weighting          : term frequency (tf)
> inspect(FICcldtm)
<<DocumentTermMatrix (documents: 1, terms: 8827)>>
Non-/sparse entries: 8827/0
Sparsity           : 0%
Maximal term length: 21
Weighting          : term frequency (tf)
Sample             :
       Terms
Docs     and but had not that  the this  was which with
  a.txt 2573 680 623 915 1024 8413  747 1326   777  864
```

**Since we only have 1 document in corpus, the sparsity is 0%. After the cleansing is done, we did the finding the 10 longest words again:**

```
> countwords<-sapply(FICcldtm$dimnames$Terms, nchar)
> dt<-data.frame(word = unlist(FICcldtm$dimnames$Terms), length = unlist(countwords))
> dt<-dt[order(dt$length),]
> tail(dt, 10)
                                      word length
petromyzonspricka         petromyzonspricka     17
compagnienationale       compagnienationale     18
emperorholocanthus       emperorholocanthus     18
harpoonercommander       harpoonercommander     18
waterstuberculated       waterstuberculated     18
communicationrather     communicationrather     19
mohammedbenabdallah     mohammedbenabdallah     19
observedturritellas     observedturritellas     19
yescertainlyperhaps     yescertainlyperhaps     19
selfconfidencebecause selfconfidencebecause     21
```

**The dashes have disappeared, we got some relatively long words here. We have the correct length of them by disregarding the dashes.**

**Corpus cleansing:**

**However, the corpus cleansing has not finished yet. Now, we remove stopwords.**

```
> mystopwords<-c(tm::stopwords('english'))
> mystopwords
  [1] "i"          "me"         "my"         "myself"     "we"         "our"        "ours"       "ourselves"  "you"
 [10] "your"       "yours"      "yourself"   "yourselves" "he"         "him"        "his"        "himself"    "she"
 [19] "her"        "hers"       "herself"    "it"         "its"        "itself"     "they"       "them"       "their"
 [28] "theirs"     "themselves" "what"       "which"      "who"        "whom"       "this"       "that"       "these"
 [37] "those"      "am"         "is"         "are"        "was"        "were"       "be"         "been"       "being"
 [46] "have"       "has"        "had"        "having"     "do"         "does"       "did"        "doing"      "would"
 [55] "should"     "could"      "ought"      "i'm"        "you're"     "he's"       "she's"      "it's"       "we're"
 [64] "they're"    "i've"       "you've"     "we've"      "they've"    "i'd"        "you'd"      "he'd"       "she'd"
 [73] "we'd"       "they'd"     "i'll"       "you'll"     "he'll"      "she'll"     "we'll"      "they'll"    "isn't"
 [82] "aren't"     "wasn't"     "weren't"    "hasn't"     "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
 [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"     "couldn't"   "mustn't"    "let's"
[100] "that's"     "who's"      "what's"     "here's"     "there's"    "when's"     "where's"    "why's"      "how's"
[109] "a"          "an"         "the"        "and"        "but"        "if"         "or"         "because"    "as"
[118] "until"      "while"      "of"         "at"         "by"         "for"        "with"       "about"      "against"
[127] "between"    "into"       "through"    "during"     "before"     "after"      "above"      "below"      "to"
[136] "from"       "up"         "down"       "in"         "out"        "on"         "off"        "over"       "under"
[145] "again"      "further"    "then"       "once"       "here"       "there"      "when"       "where"      "why"
[154] "how"        "all"        "any"        "both"       "each"       "few"        "more"       "most"       "other"
[163] "some"       "such"       "no"         "nor"        "not"        "only"       "own"        "same"       "so"
[172] "than"       "too"        "very"
```

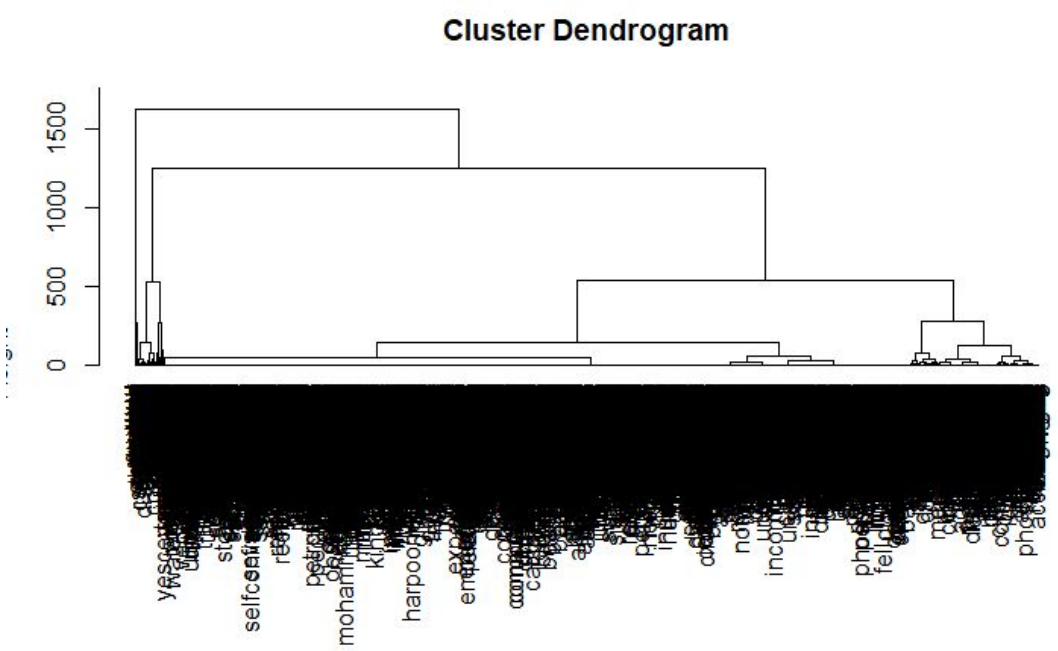
```
> FICstop<-tm_map(FICcl,tm::removeWords, mystopwords)
> FICstopdtm<-DocumentTermMatrix(FICstop)
> FICstopdtm
<<DocumentTermMatrix (documents: 1, terms: 8729)>>
Non-/sparse entries: 8729/0
Sparsity           : 0%
Maximal term length: 21
Weighting          : term frequency (tf)
```

**Dendrogram:**

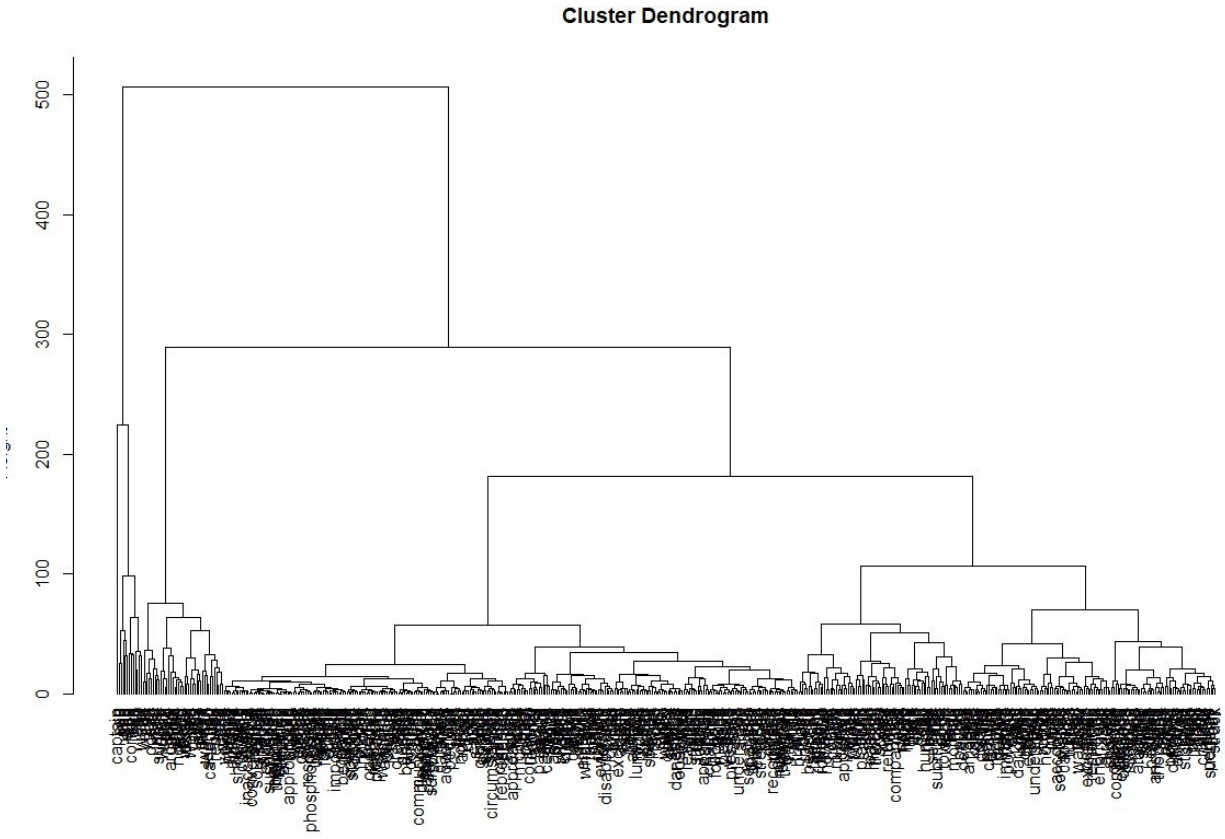
**Now the cleansing is done, below is the code we used to process the data and display a dendrogram:**

```
> dtmss <- removeSparseTerms(FICstopdtm, 0.15)
> distance<-dist(t(dtmss), method = "euclidian")
> fit<-hclust(d=distance, method = "ward.D2")
> plot(fit, hang=-1)
```

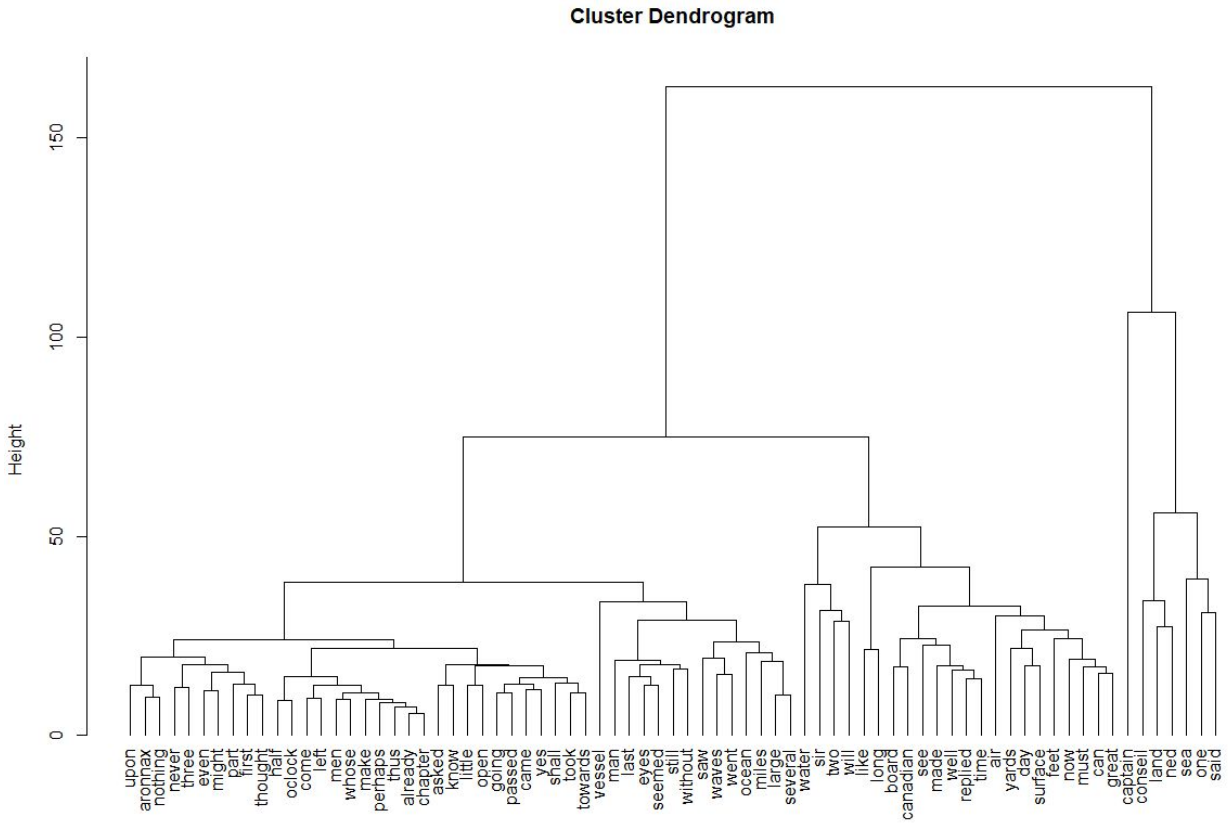
**And we get:**




**However, removing words by their sparsites is not working because we only have one document here. Therefore, we tried divide the document into several parts and do the processing again:**



**This time is much clearer than the last one. But there are still many words overlapping with each other. So, we divide the files into 30 parts and draw the dendrogram again:**

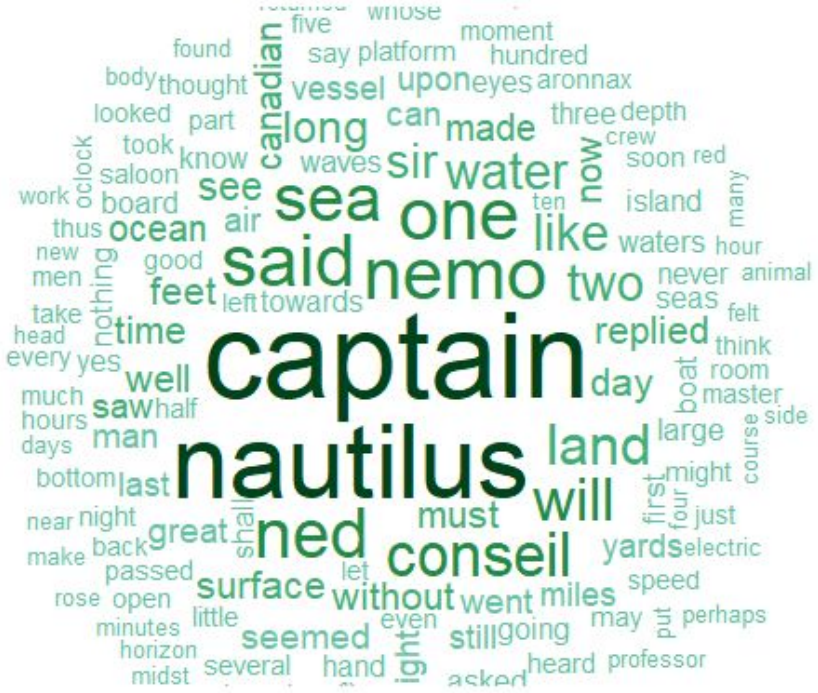



**Wordcloud:**

**Because a lot of words' frequencies are over 10, we set the min.freq = 50 in case the Word Cloud is too crowded**

```
> FICstoptdm<-TermDocumentMatrix(FICstop)
> FICstoptf<-rowSums(as.matrix(FICstoptdm))
> FICstopwf<-sort(FICstoptf, decreasing = T)
> pal<-brewer.pal(9, "BuGn")
> pal<-pal[-(1:4)]
> wordcloud(words = names(FICstopwf), freq = FICstopwf, min.freq = 50, random.order = F, colors = pal)
```

**As we can see in the Word Cloud below, the word 'captain', 'nemo', 'sea' and 'nautilus' are much bigger than other words. This picture describes the contents of this book very well.**




**Ten longest sentences:**

```
> library(readtext)
> library(ngram)
> dframe <- readtext::readtext("TwentyThousandLeagues.txt")
> doc.corpus <- quanteda::corpus(dframe)
> doc.tokens.sentence <- quanteda::tokens(doc.corpus, what = "sentence")
> countsentences<-(sapply(doc.tokens.sentence[[1]], wordcount))
> ds<-data.frame(word=unlist(doc.tokens.sentence[[1]] ), length=unlist(countsentences))
> ds<-ds[order(ds$length),]
> tail(ds,10)
```

1202
In the first group, the tubipores, were gorgones arranged like a fan, soft sponges of Syria, ises of the Moluccas, pennatules, an admirable virgularia of the Norwegian seas, variegated unbellulairae, alcyonariae, a whole series of madrepores, which my master Milne Edwards has so cleverly classified, amongst which I remarked some wonderful flabellinae oculinae of the Island of Bourbon, the "Neptune's car" of the Antilles, superb varieties of corals - in short, every

species of those curious polypi of which entire islands are formed, which will one day become continents.
6417
My nerves were somewhat calmer, but in my excited brain I saw over again all my existence on board the Nautilus; every incident, either happy or unfortunate, which had happened since my disappearance from the Abraham Lincoln - the submarine hunt, the Torres Straits, the savages of Papua, the running ashore, the coral cemetery, the passage of Suez, the Island of Santorin, the Cretan diver, Vigo Bay, Atlantis, the iceberg, the South Pole, the imprisonment in the ice, the fight among the poulps, the storm in the Gulf Stream, the Avenger, and the horrible scene of the vessel sunk with all her crew.
1232
Some are known to you, such as the thermometer, which gives the internal temperature of the Nautilus; the barometer, which indicates the weight of the air and foretells the changes of the weather; the hygrometer, which marks the dryness of the atmosphere; the storm-glass, the contents of which, by decomposing, announce the approach of tempests; the compass, which guides my course; the sextant, which shows the latitude by the altitude of the sun; chronometers, by which I calculate the longitude; and glasses for day and night, which I use to examine the points of the horizon, when the Nautilus rises to the surface of the waves."
1531
During their games, their bounds, while rivalling each other in beauty, brightness, and velocity, I distinguished the green labre; the banded mullet, marked by a double line of black; the round-tailed goby, of a white colour, with violet spots on the back; the Japanese scombrus, a beautiful mackerel of these seas, with a blue body and silvery head; the brilliant azurors, whose name alone defies description; some banded spares, with variegated fins of blue and yellow; the woodcocks of the seas, some specimens of which attain a yard in length; Japanese salamanders, spider lampreys, serpents six feet long, with eyes small and lively, and a huge mouth bristling with teeth; with many other species.
2953
Then, as specimens of other kinds, some ovoides, resembling an egg of a dark brown colour, marked with white bands, and without tails; diodons, real sea-porcupines, furnished with spikes, and capable of swelling in such a way as to look like cushions bristling with darts; hippocampi, common to every ocean; some pegasi with lengthened snouts, which their pectoral fins, being much elongated and formed in the shape of wings, allow, if not to fly, at least to shoot into the air; pigeon spatulae, with tails covered with many rings of shell; macrognathi with long jaws, an excellent fish, nine inches long, and bright with most agreeable colours; pale-coloured calliomores, with rugged heads; and plenty of chaetpdons, with long and tubular muzzles, which kill insects by shooting them, as from an air-gun, with a single drop of water.
5559
Amongst the cartilaginous ones, petromyzons-pricka, a sort of eel, fifteen inches long, with a greenish head, violet fins, grey-blue back, brown belly, silvered and sown with bright spots, the pupil of the eye encircled with gold - a curious animal, that the current of the Amazon had drawn to the sea, for they inhabit fresh waters - tuberculated streaks, with pointed snouts, and a long loose tail, armed with a long jagged sting; little sharks, a yard long, grey and whitish skin, and

several rows of teeth, bent back, that are generally known by the name of pantouffles; vespertilios, a kind of red isosceles triangle, half a yard long, to which pectorals are attached by fleshy prolongations that make them look like bats, but that their horny appendage, situated near the nostrils, has given them the name of sea-unicorns; lastly, some species of balistae, the curassavian, whose spots were of a brilliant gold colour, and the capriscus of clear violet, and with varying shades like a pigeon's throat.
1206
Amongst these specimens I will quote from memory only the elegant royal hammer-fish of the Indian Ocean, whose regular white spots stood out brightly on a red and brown ground, an imperial spondyle, bright-coloured, bristling with spines, a rare specimen in the European museums - (I estimated its value at not less than L1000); a common hammer-fish of the seas of New Holland, which is only procured with difficulty; exotic buccardia of Senegal; fragile white bivalve shells, which a breath might shatter like a soap-bubble; several varieties of the aspirgillum of Java, a kind of calcareous tube, edged with leafy folds, and much debated by amateurs; a whole series of trochi, some a greenish-yellow, found in the American seas, others a reddish-brown, natives of Australian waters; others from the Gulf of Mexico, remarkable for their imbricated shell; stellari found in the Southern Seas; and last, the rarest of all, the magnificent spur of New Zealand; and every description of delicate and fragile shells to which science has given appropriate names.
5560
I end here this catalogue, which is somewhat dry perhaps, but very exact, with a series of bony fish that I observed in passing belonging to the apteronotes, and whose snout is white as snow, the body of a beautiful black, marked with a very long loose fleshy strip; odontognathes, armed with spikes; sardines nine inches long, glittering with a bright silver light; a species of mackerel provided with two anal fins; centronotes of a blackish tint, that are fished for with torches, long fish, two yards in length, with fat flesh, white and firm, which, when they arc fresh, taste like eel, and when dry, like smoked salmon; labres, half red, covered with scales only at the bottom of the dorsal and anal fins; chrysoptera, on which gold and silver blend their brightness with that of the ruby and topaz; golden-tailed spares, the flesh of which is extremely delicate, and whose phosphorescent properties betray them in the midst of the waters; orange-coloured spares with long tongues; maigres, with gold caudal fins, dark thorn-tails, anableps of Surinam, etc. Notwithstanding this "et cetera," I must not omit to mention fish that Conseil will long remember, and with good reason.

| index | length |
|---|---|
| 1202 | 89 |
| 3075 | 91 |
| 4908 | 93 |
| 6417 | 103 |
| 1232 | 106 |
| 1531 | 114 |
| 2953 | 137 |
| 5559 | 168 |

```
1206  170
5560  198
```

```
> longest_sentences<-tail(ds,10)
> s<-as.String(longest_sentences$word)
> sent_token_annotator <- Maxent_Sent_Token_Annotator()
> word_token_annotator <- Maxent_Word_Token_Annotator()
> a2 <- annotate(s, list(sent_token_annotator, word_token_annotator))
> pos_tag_annotator <- Maxent_POS_Tag_Annotator()
> pos_tag_annotator
An annotator inheriting from classes
  Simple_POS_Tag_Annotator Annotator
with description
  Computes POS tag annotations using the Apache OpenNLP Maxent Part of
  Speech tagger employing the default model for language 'en'
```

```
> a3 <- annotate(s, pos_tag_annotator, a2)
> head(a3,13)
 id type     start end  features
  1 sentence     1  575 constituents=<<integer,108>>
  2 sentence   577 1082 constituents=<<integer,112>>
  3 sentence  1084 1599 constituents=<<integer,108>>
  4 sentence  1601 2201 constituents=<<integer,124>>
  5 sentence  2203 2836 constituents=<<integer,127>>
  6 sentence  2838 3539 constituents=<<integer,140>>
  7 sentence  3541 4375 constituents=<<integer,168>>
  8 sentence  4377 5383 constituents=<<integer,203>>
  9 sentence  5385 6441 constituents=<<integer,199>>
 10 sentence  6443 7627 constituents=<<integer,239>>
 11 word         1    2 POS=IN
 12 word         4    6 POS=DT
 13 word         8   12 POS=JJ
```

```
> head(annotate(s, Maxent_POS_Tag_Annotator(probs = TRUE), a2))
 id type     start end  features
  1 sentence     1  575 constituents=<<integer,108>>
  2 sentence   577 1082 constituents=<<integer,112>>
  3 sentence  1084 1599 constituents=<<integer,108>>
  4 sentence  1601 2201 constituents=<<integer,124>>
  5 sentence  2203 2836 constituents=<<integer,127>>
  6 sentence  2838 3539 constituents=<<integer,140>>
> a3w <- subset(a3, type == "word")
> head(a3w)
 id type start end features
 11 word     1   2 POS=IN
 12 word     4   6 POS=DT
 13 word     8  12 POS=JJ
 14 word    14  18 POS=NN
 15 word    19  19 POS=,
 16 word    21  23 POS=DT
```

```
> a4w <- subset(a3w, type == "word" & end-start > 4  )
> tags <- sapply(a4w$features, `[[`, "POS")
> verbs_great_six=a4w[sapply(tags, function(x) { grepl("^VB.*|NN*", x)  })]
> head(verbs_great_six,12)
 id type start end features
 17 word    25  33 POS=NNS
 20 word    41  48 POS=NNS
 21 word    50  57 POS=VBN
 27 word    76  82 POS=NNS
 34 word   106 113 POS=NNP
 36 word   116 125 POS=NNS
 40 word   141 150 POS=NN
 46 word   175 184 POS=VBN
 47 word   186 198 POS=NNS
 49 word   201 211 POS=NNS
 53 word   222 227 POS=NN
 55 word   232 241 POS=NNS
>
```

```
> sprintf("%s", s[verbs_great_six])
 [1] "tubipores"      "gorgones"       "arranged"       "sponges"
 [5] "Moluccas"       "pennatules"     "virgularia"     "variegated"
 [9] "unbellulairae"  "alcyonariae"    "series"         "madrepores"
[13] "master"         "Edwards"        "classified"     "amongst"
[17] "remarked"       "flabellinae"    "oculinae"       "Island"
[21] "Bourbon"        "Neptune"        "Antilles"       "varieties"
```

```
 [25] "corals"             "species"           "polypi"            "islands"
 [29] "formed"             "become"            "continents"        "answered"
 [33] "Orientals"          "solidified"        "ladies"            "brilliancy"
 [37] "mother-of-pearl"    "substance"         "fingers"           "chemist"
 [41] "mixture"            "phosphate"         "carbonate"         "gelatine"
 [45] "naturalists"        "morbid"            "secretion"         "produces"
 [49] "mother-of-pearl"    "amongst"           "bivalves"          "Albatrosses"
 [53] "passed"             "expanse"           "called"            "vultures"
 [57] "petrels"            "damiers"           "underpart"         "series"
 [61] "petrels"            "others"            "Antarctic"         "Conseil"
 [65] "inhabitants"        "Ferroe"            "Islands"           "nothing"
 [69] "lighting"           "nerves"            "existence"         "Nautilus"
 [73] "incident"           "happened"          "disappearance"     "Abraham"
 [77] "Lincoln"            "submarine"         "Torres"            "Straits"
 [81] "savages"            "running"           "cemetery"          "passage"
 [85] "Island"             "Santorin"          "Cretan"            "Atlantis"
 [89] "iceberg"            "imprisonment"      "poulps"            "Stream"
 [93] "Avenger"            "vessel"            "thermometer"       "temperature"
 [97] "Nautilus"           "barometer"         "indicates"         "weight"
[101] "foretells"          "changes"           "weather"           "hygrometer"
[105] "dryness"            "atmosphere"        "storm-glass"       "contents"
[109] "decomposing"        "announce"          "approach"          "tempests"
[113] "compass"            "guides"            "course"            "sextant"
[117] "latitude"           "altitude"          "chronometers"      "calculate"
[121] "longitude"          "glasses"           "examine"           "points"
[125] "horizon"            "Nautilus"          "surface"           "During"
[129] "bounds"             "rivalling"         "beauty"            "brightness"
[133] "velocity"           "distinguished"     "banded"            "mullet"
[137] "marked"             "colour"            "violet"            "scombrus"
[141] "mackerel"           "azurors"           "defies"            "description"
[145] "banded"             "spares"            "woodcocks"         "specimens"
[149] "attain"             "length"            "salamanders"       "spider"
[153] "lampreys"           "serpents"          "bristling"         "species"
[157] "specimens"          "ovoides"           "resembling"        "colour"
[161] "marked"             "without"           "diodons"           "sea-porcupines"
[165] "furnished"          "spikes"            "swelling"          "cushions"
[169] "bristling"          "hippocampi"        "pegasi"            "lengthened"
[173] "snouts"             "formed"            "pigeon"            "spatulae"
[177] "covered"            "macrognathi"       "inches"            "colours"
[181] "calliomores"        "plenty"            "chaetpdons"        "muzzles"
[185] "insects"            "shooting"          "air-gun"           "Amongst"
[189] "petromyzons-pricka" "inches"            "violet"            "encircled"
[193] "animal"             "Amazon"            "inhabit"           "waters"
```

```
[197] "streaks"          "snouts"           "sharks"           "pantouffles"
[201] "vespertilios"     "isosceles"        "triangle"         "pectorals"
[205] "attached"         "prolongations"    "appendage"        "situated"
[209] "nostrils"         "sea-unicorns"     "species"          "balistae"
[213] "colour"           "capriscus"        "violet"           "varying"
[217] "shades"           "pigeon"           "throat"           "Amongst"
[221] "specimens"        "memory"           "hammer-fish"      "Indian"
[225] "ground"           "spondyle"         "bristling"        "spines"
[229] "specimen"         "museums"          "estimated"        "hammer-fish"
[233] "Holland"          "procured"         "difficulty"       "buccardia"
[237] "Senegal"          "bivalve"          "shells"           "breath"
[241] "shatter"          "soap-bubble"      "varieties"        "aspirgillum"
[245] "debated"          "amateurs"         "series"           "trochi"
[249] "American"         "others"           "natives"          "waters"
[253] "others"           "Mexico"           "rarest"           "Zealand"
[257] "description"      "shells"           "science"          "catalogue"
[261] "series"           "observed"         "passing"          "belonging"
[265] "apteronotes"      "marked"           "fleshy"           "odontognathes"
[269] "spikes"           "sardines"         "inches"           "glittering"
[273] "species"          "mackerel"         "provided"         "centronotes"
[277] "fished"           "torches"          "length"           "smoked"
[281] "salmon"           "labres"           "covered"          "scales"
[285] "bottom"           "chrysoptera"      "silver"           "brightness"
[289] "spares"           "properties"       "betray"           "waters"
[293] "spares"           "tongues"          "maigres"          "caudal"
[297] "thorn-tails"      "anableps"         "Surinam"          "Notwithstanding"
[301] "cetera"           "mention"
```

```
> library(RWeka)
> BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
> TrigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 3, max = 3))
```

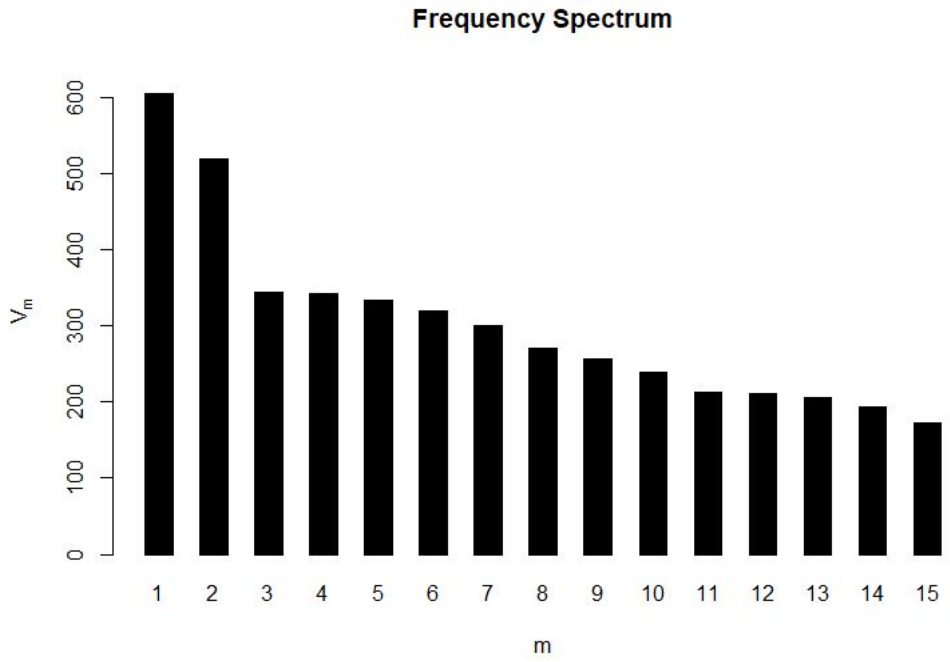
```
> dir.create("../new")
> file.create("../new/foo.txt")
[1] TRUE
> sent<-paste(sent, collapse = " ")
> writeLines(sent, fileConn)
> corp<-VCorpus(DirSource("../new",ignore.case = TRUE,mode = "text"))
> trigram<- tm::TermDocumentMatrix(corp, control = list(tokenize = TrigramTokenizer))
> trigrams_used<-trigram$dimnames$Terms
> head(trigrams_used)
[1] "abraham lincoln submarine"          "air-gun amongst petromyzons-pricka"
[3] "albatrosses passed expanse"         "alcyonariae series madrepores"
[5] "altitude chronometers calculate"    "amateurs series trochi"
```

```
> bigram <- tm::TermDocumentMatrix(corp, control = list(tokenize = BigramTokenizer))
> bigram_used <- bigram$dimnames$Terms
> head(bigram_used,20)
 [1] "abraham lincoln"            "air-gun amongst"
 [3] "albatrosses passed"         "alcyonariae series"
 [5] "altitude chronometers"      "amateurs series"
 [7] "amazon inhabit"             "american others"
 [9] "amongst bivalves"           "amongst petromyzons-pricka"
[11] "amongst remarked"           "amongst specimens"
[13] "anableps surinam"           "animal amazon"
[15] "announce approach"          "answered orientals"
[17] "antarctic conseil"          "antilles varieties"
[19] "appendage situated"         "approach tempests"
>
```

**Using zipfR to analyze the word frequency:**

```
library(zipfR)
library(stringr)
FICstopwf.spc<-spc(FICstopwf, 1:length(FICstopwf))
plot(FICstopwf.spc)
```

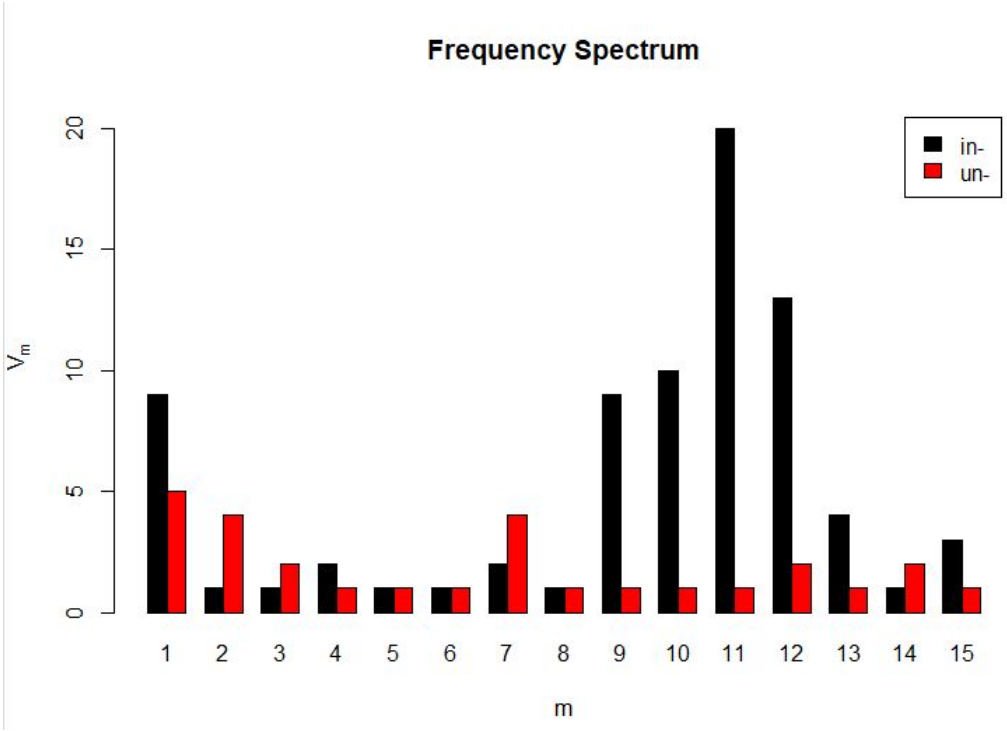
**First we import the package. Then, we plot the frequency spectrum of some words with the largest frequency:**




**Then, to further explore the text. We try to compare two prefix ‘in’ and ‘un’ in english to see which one is more productive**

```
words.in<-startsWith(FICstopdtm$dimnames$Terms, 'in')
words.in<-FICstoptf[which(words.in == TRUE)]
words.un<-startsWith(FICstopdtm$dimnames$Terms, 'un')
words.un<-FICstoptf[which(words.un == TRUE)]
words.in.spc<-spc(words.in, 1:length(words.in))
words.un.spc<-spc(words.un, 1:length(words.un))
plot(words.in.spc,words.un.spc,legend=c("in-","un-"))
words.un.fzm <- lnre("fzm",words.un.spc)
summary(words.un.fzm)
words.un.ext.spc<-lnre.spc(words.un.fzm, N(words.in.spc))
Vm(words.un.ext.spc,1)/N(words.in.spc)
Vm(words.in.spc,1)/N(words.in.spc)
sample.sizes <- floor(N(words.in.spc)/100)*(1:100)
words.in.vgc <- vgc.interp(words.in.spc, sample.sizes)
words.un.vgc <- lnre.vgc(words.un.fzm, sample.sizes)
plot(words.in.vgc,words.un.vgc,legend=c("in-","un-"))
```

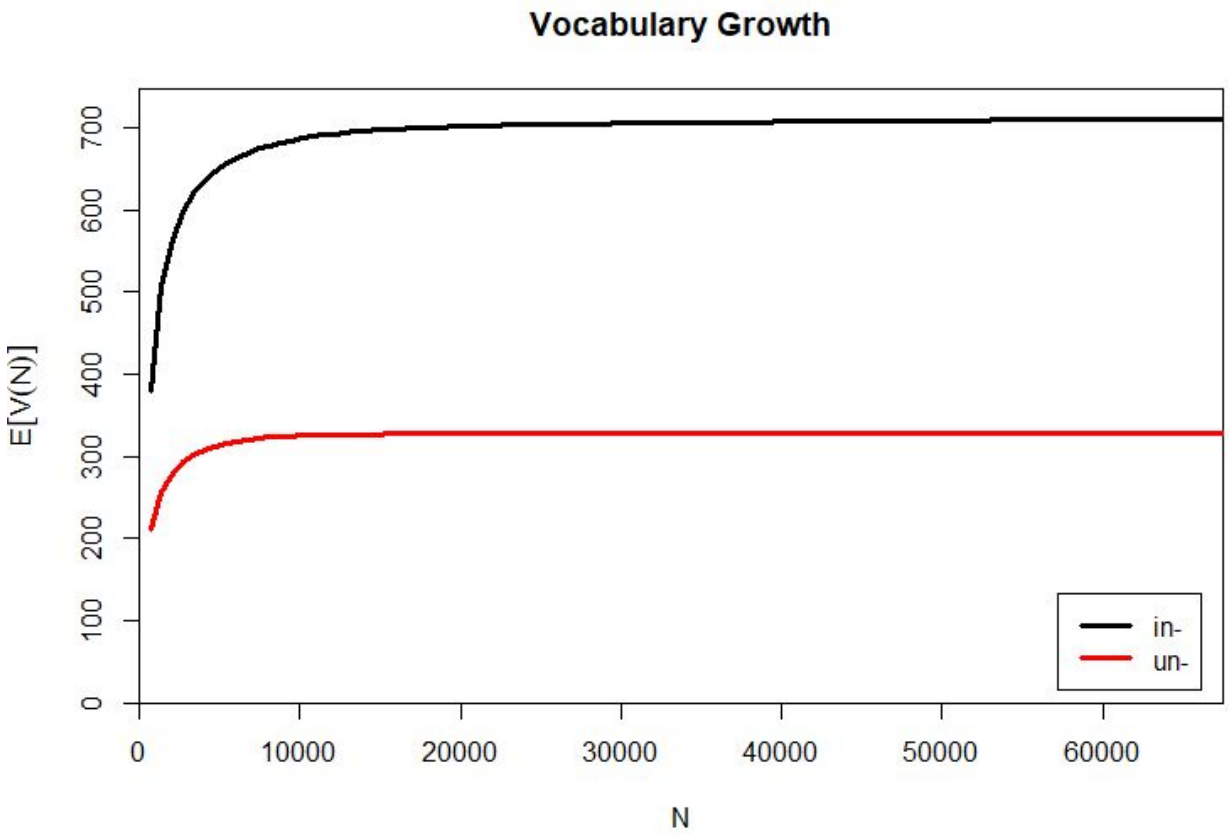
**Above is the code we use. Below we output two plots to show the result.**




```
> summary(words.un.fzm)
finite Zipf-Mandelbrot LNRE model.
Parameters:
   Shape:          alpha = 7.838051e-07
   Lower cutoff:       A = 0.0002364176
   Upper cutoff:       B = 0.01231131
 [ Normalization:      C = 82.8161 ]
Population size: S = 327.3467
Sampling method: Poisson, with exact calculations.

Parameters estimated from sample of size N = 17917:
                  V  V1   V2   V3   V4   V5
   Observed: 327.00 5.0 4.00 2.00 1.00 1.00 ...
   Expected: 327.11 1.2 3.14 5.67 8.05 9.65 ...

Goodness-of-fit (multivariate chi-squared test):
         X2 df            p
   34.51919  3 1.539213e-07
> words.un.ext.spc<-lnre.spc(words.un.fzm, N(words.in.spc))
> Vm(words.un.ext.spc,1)/N(words.in.spc)
[1] 1.431009e-10
> Vm(words.in.spc,1)/N(words.in.spc)
[1] 0.0001332859
> sample.sizes <- floor(N(words.in.spc)/100)*(1:100)
> words.in.vgc <- vgc.interp(words.in.spc, sample.sizes)
> words.un.vgc <- lnre.vgc(words.un.fzm, sample.sizes)
> plot(words.in.vgc,words.un.vgc,legend=c("in-","un-"))
```




**As we can see the graph above, when the sample size is smaller than 10000, the prefix 'in' is more productive for the vocabulary size. After 10000, the size remains nearly unchanged.**

**For other R packages for text processing, we tried some functions.**
**Stringi:**
**To use the functions in Stringi, we extract 1 sentence out of the document:**

```
> test1$content[15]
[1] "The year 1866 was signalised by a remarkable incident, a mysterious and"
> test1string<-test1$content[15]
> stri_replace_all(test1string, "", regex = "a")
[1] "The yer 1866 ws signlised by  remrkble incident,  mysterious nd"
```

**And we use the function stri_replace_all to replace all the character 'a' with '' in this sentence. Above is the result. Also, we tried to remove any words starting with 'in':**

```
> stri_replace_all(test1string, "", regex = "in[A-Za-z]+")
[1] "The year 1866 was signalised by a remarkable , a mysterious and"
```

**As we can see, the word 'incident' is removed.**

```
> test1string<-"   The year 1866 was signalised by a remarkable incident, a mysterious and      "
> stri_trim(test1string)
[1] "The year 1866 was signalised by a remarkable incident, a mysterious and"
```

**Then, we used the function stri_trim(). We added some spaces at the beginning and the end of this sentence to see how the function works.**

```
> test1string<-"   The year 1866 was signalised by a remarkable incident, a MYSTERIOUS and      "
> stri_trans_tolower(test1string)
[1] "   the year 1866 was signalised by a remarkable incident, a mysterious and      "
```

**The function stri_trans_tolower() also works fine.**

**Quanteda:**

**Using the function corpus() to create a new corpus:**

```
> dataframe <- readtext("a.txt",encoding = "UTF-8")
> unlink("tmp", recursive = TRUE)
> doc.corpus <- corpus(dataframe)
> summary(doc.corpus)
Corpus consisting of 1 document, showing 1 document:

  Text Types Tokens Sentences
 a.txt  9701 123526      6586
```

**Tokenization with function tokens()**

```
> doc.tokens <- tokens(doc.corpus)
> summary(doc.tokens)
      Length Class  Mode
a.txt 123526 -none- character
> doc.tokens
Tokens consisting of 1 document.
a.txt :
 [1] "TWENTY"   "THOUSAND" "LEAGUES"  "UNDER"    "THE"      "SEA"      "by"       "JULES"    "VERNE"    "PART"     "ONE"
[12] "CHAPTER"
[ ... and 123,514 more ]
```

**Tokenization with function tokens_select() to remove stop words in tokens**

```
> doc.tokens <- tokens_select(doc.tokens, stopwords('english'),selection='remove')
> doc.tokens
Tokens consisting of 1 document.
a.txt :
 [1] "TWENTY"   "THOUSAND" "LEAGUES"  "SEA"      "JULES"    "VERNE"    "PART"     "ONE"      "CHAPTER"  "SHIFTING" "REEF"
[12] "year"
[ ... and 71,466 more ]
```

**The size decreased from 123,514 to 71,466.**

**Extract sentences using function tokens() with parameter what = “sentence”**

```
> doc.tokens.sentence <- tokens(doc.corpus, what = "sentence")
> doc.tokens.sentence
Tokens consisting of 1 document.
a.txt :
 [1] " TWENTY THOUSAND LEAGUES UNDER THE SEA  by  JULES VERNE   PART ONE  CHAPTER I  A SHIFTING REEF  The year 1866 was signalised by a r
emarkable incident, a mysterious and puzzling phenomenon, which doubtless no one has yet forgotten."

 [2] "Not to mention rumours which agitated the maritime population and excited the public mind, even in the interior of continents, seaf
aring men were particularly excited."

 [3] "Merchants, common sailors, captains of vessels, skippers, both of Europe and America, naval officers of all countries, and the Gove
rnments of several States on the two continents, were deeply interested in the matter."

 [4] "For some time past vessels had been met by \"an enormous thing,\" a long object, spindle-shaped, occasionally phosphorescent, and i
nfinitely larger and more rapid in its movements than a whale."
```

**Create a document feature matrix(dfm) with function dfm()**

```
> doc.dfm.final <- dfm(doc.tokens)
> doc.dfm.final
Document-feature matrix of: 1 document, 8,911 features (0.0% sparse).
       features
docs    twenty thousand leagues sea jules verne part one chapter shifting
  a.txt     44       39      31 302     1     1   81 348      46        2
[ reached max_nfeat ... 8,901 more features ]
```

**Getting top features:**

```
> topfeatures(doc.dfm.final, 5)
   ,    .    "    ;    ?
8826 5640 3404  840  663
```

**Since we did not remove punctuations here, the top 5 features are not words.**

**Tidytext:**

**Tidy a corpus from the tm package:**

```
> tidy(FIC, collapse = NULL)
# A tibble: 30 x 8
   author datetimestamp       description heading id       language origin text
   <lgl>  <dttm>              <lgl>       <lgl>   <chr>    <chr>    <lgl>  <named list>
 1 NA     2020-04-30 15:02:10 NA          NA      a_1.txt  en       NA     <chr [405]>
 2 NA     2020-04-30 15:02:10 NA          NA      a_10.txt en       NA     <chr [405]>
 3 NA     2020-04-30 15:02:10 NA          NA      a_11.txt en       NA     <chr [405]>
 4 NA     2020-04-30 15:02:10 NA          NA      a_12.txt en       NA     <chr [405]>
 5 NA     2020-04-30 15:02:10 NA          NA      a_13.txt en       NA     <chr [405]>
 6 NA     2020-04-30 15:02:10 NA          NA      a_14.txt en       NA     <chr [405]>
 7 NA     2020-04-30 15:02:10 NA          NA      a_15.txt en       NA     <chr [405]>
 8 NA     2020-04-30 15:02:10 NA          NA      a_16.txt en       NA     <chr [405]>
 9 NA     2020-04-30 15:02:10 NA          NA      a_17.txt en       NA     <chr [405]>
10 NA     2020-04-30 15:02:10 NA          NA      a_18.txt en       NA     <chr [405]>
# ... with 20 more rows
```

**Get stopwords:**

```
> get_stopwords(language = "en", source = "snowball")
# A tibble: 175 x 2
   word      lexicon
   <chr>     <chr>
 1 i         snowball
 2 me        snowball
 3 my        snowball
 4 myself    snowball
 5 we        snowball
 6 our       snowball
 7 ours      snowball
 8 ourselves snowball
 9 you       snowball
10 your      snowball
# ... with 165 more rows
```

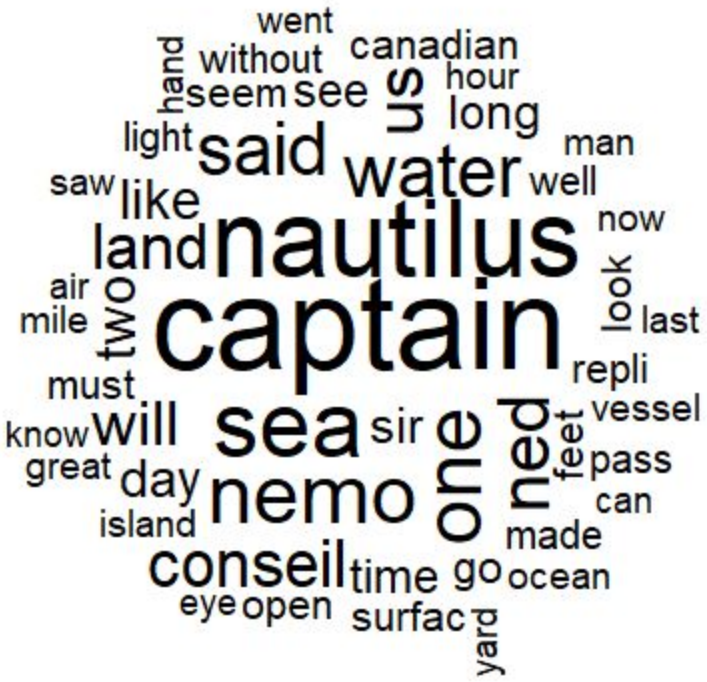
**Tidy a DocumentTermMatrix or TermDocumentMatrix into a three-column data frame:**

```
> tidy(FICstopdtm)
# A tibble: 29,424 x 3
   document term        count
   <chr>    <chr>       <dbl>
 1 a_1.txt  abandoned       1
 2 a_1.txt  abraham         3
 3 a_1.txt  absence         1
 4 a_1.txt  abyss           1
 5 a_1.txt  accepted        1
 6 a_1.txt  accident        3
 7 a_1.txt  accompanied     1
 8 a_1.txt  according       1
 9 a_1.txt  accounted       1
10 a_1.txt  accused         1
# ... with 29,414 more rows
> tidy(FICstoptdm)
# A tibble: 29,424 x 3
   term        document count
   <chr>       <chr>    <dbl>
 1 abandoned   a_1.txt      1
 2 abraham     a_1.txt      3
 3 absence     a_1.txt      1
 4 abyss       a_1.txt      1
 5 accepted    a_1.txt      1
 6 accident    a_1.txt      3
 7 accompanied a_1.txt      1
 8 according   a_1.txt      1
 9 accounted   a_1.txt      1
10 accused     a_1.txt      1
# ... with 29,414 more rows
```

**Corpustool:**

**We create a new corpus using create_tcorpus() using the older one's content. Then we did the preprocessing to remove the stop words and get the document term matrix by using function preprocess() and dtm(). Last but not least, we create a Word Cloud using function dtm_wordcloud()**

```
> FIC_tc<-VCorpus(DirSource(".", ignore.case = TRUE, mode = "text"))
> tc<-create_tcorpus(FIC_tc[[1]]$content)
  |=================================================================================
> tc$preprocess('token', 'stem', remove_stopwords = TRUE, use_stemming = TRUE)
> dtm<-tc$dtm(feature = 'stem')
> dtm_wordcloud(dtm, nterms = 50)
```




**The word cloud is quite similar to the word cloud that is created by applying functions in the package 'wordcloud'.**

**Functions we wrote:**
**This is used for removing all the numbers and punctuations:**
**removeNumPunct<-function(x) gsub("[^[:alpha:][:space:]*]", "", x)**
**These two is used for getting the bigram and trigram:**

```
BigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
TrigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 3, max = 3))
```

**What have we learnt:**
**To do the text processing, we have to do the cleansing first. After building the corpus, we have to remove the punctuation and Stopwords in documents. Moreover, there might be some words that do not appear in other documents which will increase the sparsity. We have to remove them, too. After that, we can do the tokenization. Then, we can get word frequency and go further. The package 'zipfR' is interesting. We find all the words starting with 'in' and 'un' and compare them. Seems that if a person knows more words starting with 'in' indicates that he or she has bigger vocabulary than knowing words starting with 'un'.**

**Furthermore, ‘plotting is your friend’ as it is said, the word cloud is a powerful tool for data visualization. It basically reflects what the text is mainly about. People can get the information directly.**

**Through learning how to make use of the NLP package related to Part Of Speech (POS) analysis; a deep understanding was gained. Not only was there insight into how POS was associated with words but also an overall intuition was obtained about what it meant to do POS analysis. Using WordNet and NLP packages we were able to generate a POS description for each word that included its start, end and what type of POS that the word was most probable to be. In doing so, we were able to single out Noun and Verb types of certain length to be returned as output.**
**Additionally an exploration of those Nouns and Verbs was done to show how bigrams and trigrams could be generated using them.**