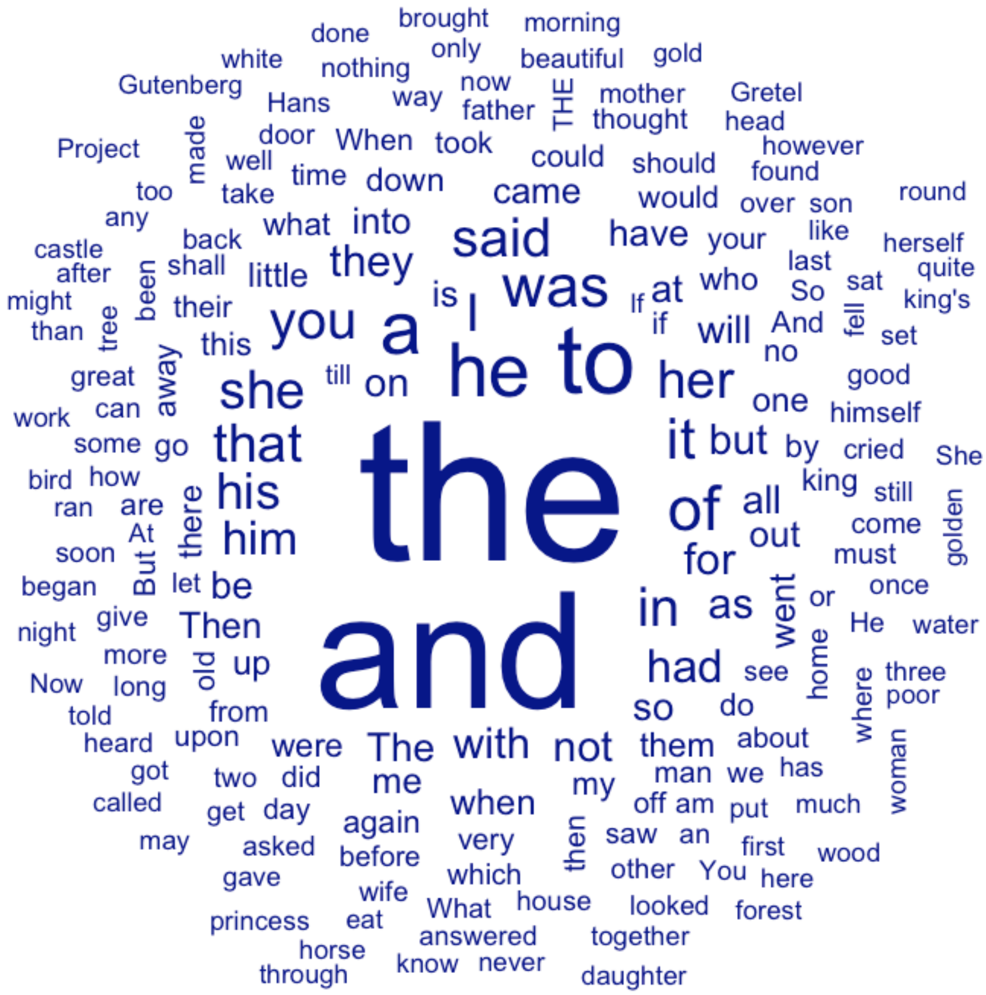
Results and Interpretation

Data of *Grimms' Fairy Tale*

```
> docnames(grimms.corpus)<-c("Grimms' Fairy Tale")
> summary(grimms.corpus)
Corpus consisting of 1 document:

                Text Types Tokens Sentences      party
 Grimms' Fairy Tale  6066 124954      4691 2591-0.txt
```

the *Grimms' Fairy Tale* has 124954 tokens and 4691 sentences, and all the tokens can be grouped into 6066 types, and they all belongs to the document of 2591-0.txt.



From the word cloud picture, "the", "to" and "and" are the most frequent words in the document, and the words ("she", "you", "he", "her", "his", and "him") are also obvious, which indicates that all the tales in the book can't be told without story characters, besides, "was", "said", and "came", these past tense words also indicate that the tales was already happened and it should be told to someone.

```
> head(grimms_kw, 10)

 [Grimms' Fairy Tale, 196] TURNIP CLEVER HANS THREE LANGUAGES | FOX | CAT FOUR CLEVER BROTHERS LILY
 [Grimms' Fairy Tale, 203]     FOUR CLEVER BROTHERS LILY LION | FOX | HORSE BLUE LIGHT RAVEN GOLDEN
 [Grimms' Fairy Tale, 223]   KNOWALL SEVEN RAVENS WEDDING MRS | FOX | FIRST STORY SECOND STORY SALAD
 [Grimms' Fairy Tale, 394]            came wood side wood saw | fox | sitting took bow made ready
 [Grimms' Fairy Tale, 401]          took bow made ready shoot | fox | said shoot will give good
 [Grimms' Fairy Tale, 444]       beast know matter shot arrow | fox | missed set tail back ran
 [Grimms' Fairy Tale, 497]         son set thing happened met | fox | gave good advice came two
 [Grimms' Fairy Tale, 553]            rest home came wood met | fox | heard good counsel thankful fox
 [Grimms' Fairy Tale, 558]    fox heard good counsel thankful | fox | attempt life brothers done fox
 [Grimms' Fairy Tale, 563]     fox attempt life brothers done | fox | said Sit upon tail will
```

I chose “fox” as my keyword, which I thought should stand for the smart character in fairy tales, and found the 10 sentences that included “fox”.

```
> ngram<-tokens_ngrams(grimms.tokens.st, n = 2)
> summary(ngram)
                   Length Class  Mode
Grimms' Fairy Tale 47250  -none- character
> head(ngram[[1]], 10)
 [1] "Project_Gutenberg" "Gutenberg_EBook"   "EBook_Grimms"      "Grimms_Fairy"
 [5] "Fairy_Tales"       "Tales_Brothers"    "Brothers_Grimm"    "Grimm_eBook"
 [9] "eBook_use"         "use_anyone"
```

Then I did a N-gram analysis of the data. It turned out a sentence like “project Gutenberg Ebook Grimms Fairy Tales Brothers Grimm ebook use anyone”, which showed the source information of the dataset.

```
> topfeatures(grimms_dfm_st)
  said   will   came   went    one little   away   king     go   took
  1162    577    461    445    407    401    295    278    278    247
> prop_grimms_dfm_st<- dfm_weight(grimms_dfm_st, scheme  = "prop")
> topfeatures(prop_grimms_dfm_st)
       said        will        came        went         one      little        away        king
0.024592072 0.012211382 0.009756407 0.009417790 0.008613574 0.008486593 0.006243254 0.005883473
         go        took
0.005883473 0.005227403
```

After tokenization, I shrunk the datas and retested the top feature words of it, and also plot two different word size text cloud picture of the shrunk data for double check. The top ten words of the shrunk data are: “said”, “will”, “came”, “went”, “one”, “little”, “away”, “king”, “go”, and “took”. We can also find there are so many past tense verbal appeared.

```
> tfidf_grimms_dfm_st<-dfm_tfidf(grimms_dfm_st)
> topfeatures(tfidf_grimms_dfm_st)
  project gutenberg     ebook    grimms     fairy     tales  brothers     grimm       use    anyone
        0         0         0         0         0         0         0         0         0         0
```

There is only one document on this data, so it is meaningless to calculate the TD-IDF, and the R Studio also run out “0” results.

Data of *Judge emotions about nuclear energy from Twitter*

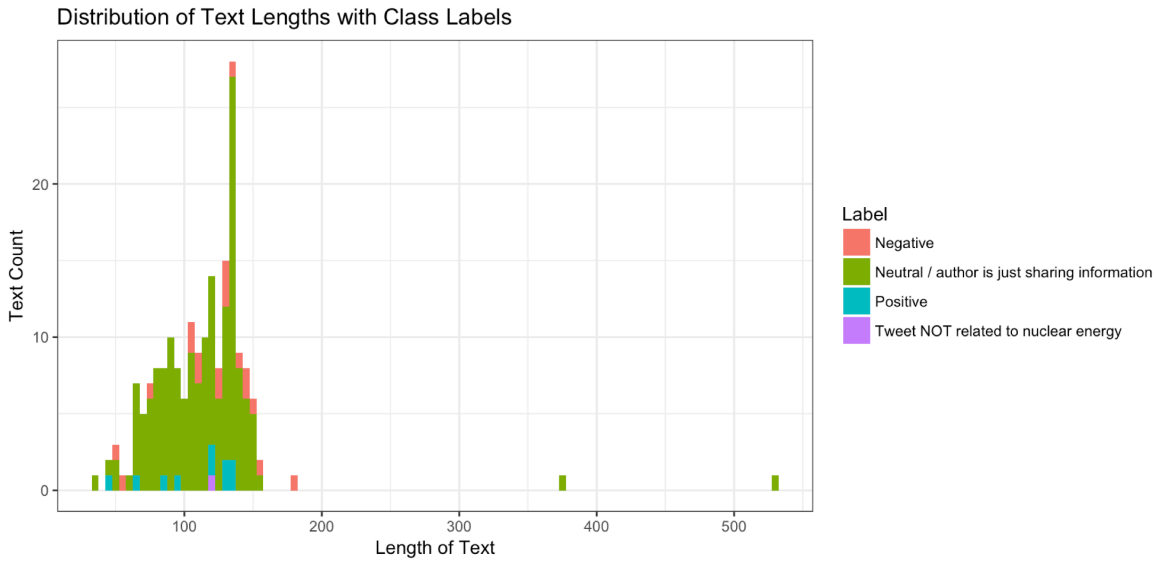
```
> table(emotion$Label)

                                Negative Neutral / author is just sharing information
                                      19                                          160
                                Positive          Tweet NOT related to nuclear energy
                                      10                                            1
> prop.table(table(emotion$Label))

                                Negative Neutral / author is just sharing information
                             0.100000000                                 0.842105263
                                Positive          Tweet NOT related to nuclear energy
                             0.052631579                                 0.005263158
>
```

From the sentiment result, the most tweets show the neutral attitude for the nuclear energy,

which means 84% tweets was just sharing the information of nuclear energy. Tweets which are negative with nuclear energy are almost twice as amount of tweets contains positive words. There is only one tweet not related to nuclear energy.

```
> summary(emotion$TextLength)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   37.0    90.0   116.5   114.3   134.0   532.0
```

Above results showed that the longest tweet contains 532 words, and the shortest tweet contains 37 words. The mean amount of words contained in the tweets is 114.



From the diagram above, the text length of most tweets are between 70 to 150. We can also double check all the results from previous work. The green occupied the most part, which has the pretty same result that 84% tweets are neutral with nuclear energy. The purple which stands for tweet not related to nuclear energy only hold one grid in this diagram, and it also match the result that there is only one tweet not related to nuclear energy. The longest and shortest texts both belong to the neutral tweets.

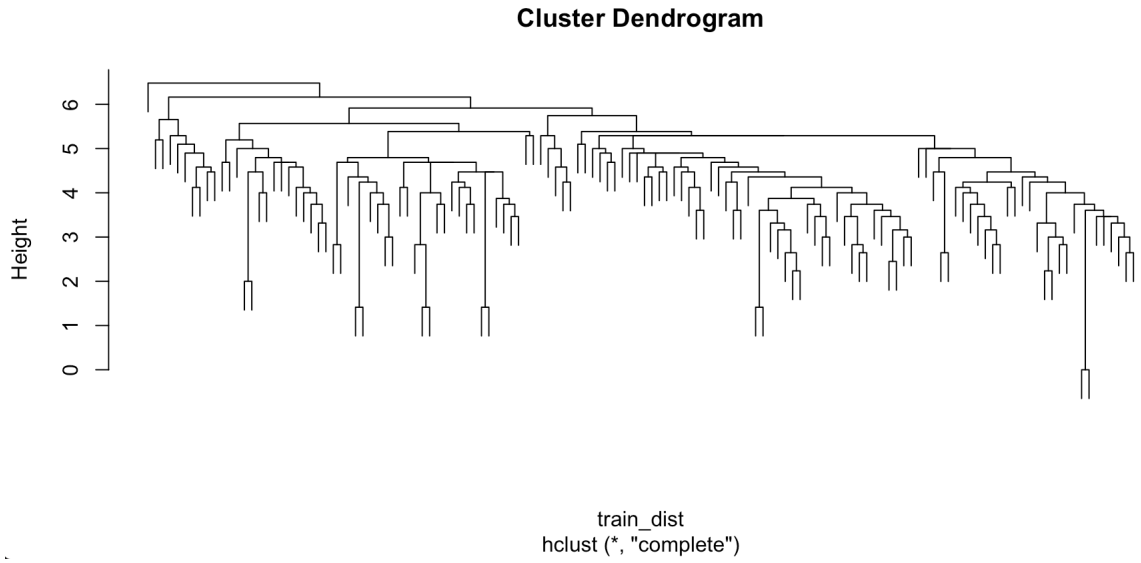
Then I split the data into a training set and a test set, and created a 0.7/0.3 stratified split.

```
> table(train$Label)

                                  Negative Neutral / author is just sharing information
                                        14                                          112
                                  Positive          Tweet NOT related to nuclear energy
                                         7                                            1
> prop.table(table(train$Label))

                                  Negative Neutral / author is just sharing information
                                0.104477612                                 0.835820896
                                  Positive          Tweet NOT related to nuclear energy
                                0.052238806                                 0.007462687
>
```

The result showed the proportion of four sentiments of the tweets in the data are almost identical with the original data. Negative tweets are as twice as positive, and neutral tweets counts for 84% in the dataset. The only one non-related tweet is spilt into this training set.

```
> table(test$Label)

                                  Negative Neutral / author is just sharing information
                                         5                                           48
                                  Positive          Tweet NOT related to nuclear energy
                                         3                                            0
> prop.table(table(test$Label))

                                  Negative Neutral / author is just sharing information
                                 0.08928571                                  0.85714286
                                  Positive          Tweet NOT related to nuclear energy
                                 0.05357143                                  0.00000000
>
```

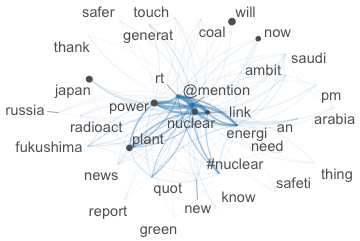
The result of test data, has the same result with training set. The proportion of different sentiment tweets also identical with the original dataset. The non-related tweet is not included in this test set.



I clustered the training set. If we cut a line in height 5.5, the training set can be basically classified into 4 groups.

```
> topfeatures(train.tokens.dfm, n=10, decreasing = TRUE)
 @mention   nuclear        rt      link     power    energi     plant  #nuclear      quot
      142       124        85        81        79        56        40        24        20
fukushima
        9
```

hack



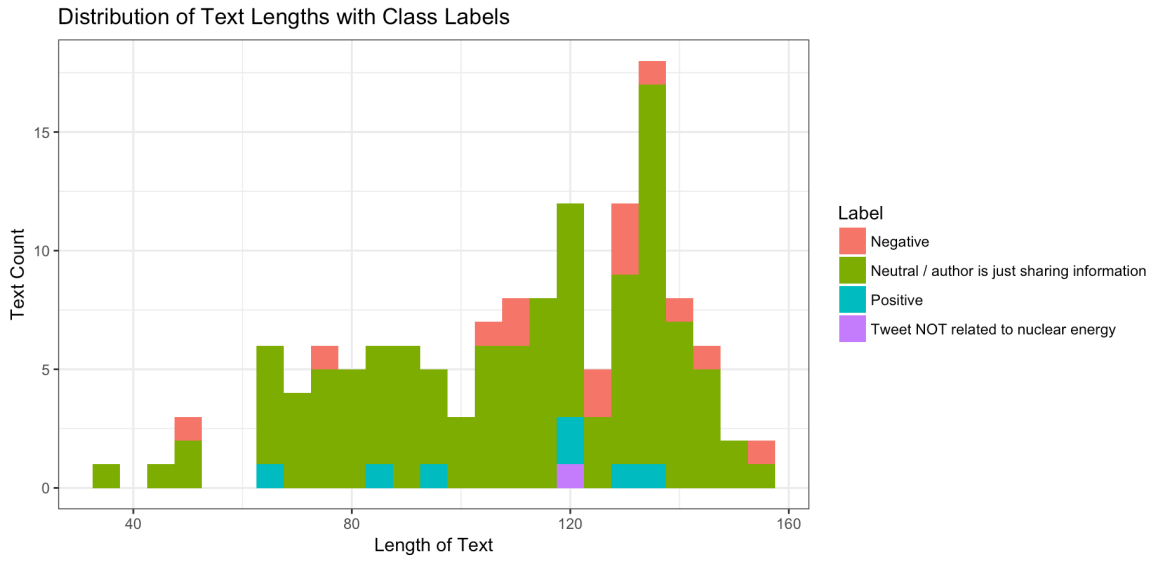
field

then I top featured the 10 top words of the training set, and visualized the network between

them. “@mention” and “nuclear” are the top words obviously, because the data is used for analyzing the sentiment of different tweets for the nuclear energy. The most amount of tweets is holding neutral position, and we can assume that they are just sharing information to others. To share information with others, they must use “@mention” to remind others, and the information is related with “nuclear energy”, so it is no doubt that “nuclear” will be the top word of the training set.

```
> summary(train$TextLength)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  37.00   89.25  116.50  110.20  133.75  155.00
>
```

From summary result of the training set, the longest text contains 155 words in the training set, and the shortest text has the same amount of words with the original dataset.



After summary, I plot training set to double check the results from the previous work. Neutral tweets still hold the most part of the chart, and the text length of most tweets is between 80 to 140.

```
> total.time
Time difference of 18.13435 secs
> rpart.cv.1
CART

134 samples
759 predictors
  4 classes: 'Negative', 'Neutral / author is just sharing information', 'Positive', 'Tweet NOT related to nuclear energy'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 119, 120, 121, 122, 120, 120, ...
Resampling results:

  Accuracy   Kappa
  0.8384188  0

Tuning parameter 'cp' was held constant at a value of 0
```

In the first training process, there are 134 observations and 759 terms, and the sentiment was divided into 4 levels: neutral; negative; positive; and non-related. The accuracy of this training set is 0.838.

```
> total.time
Time difference of 10.43279 secs
> rpart.cv.2
CART

134 samples
758 predictors
  4 classes: 'Negative', 'Neutral / author is just sharing information', 'Positive', 'Tweet NOT related to nuclear energy'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 119, 120, 121, 122, 120, 120, ...
Resampling results:

  Accuracy   Kappa
  0.8361966  -0.002898551

Tuning parameter 'cp' was held constant at a value of 0
>
```

In the second training process, there are 134 observations and 758 terms, and the sentiment still be divided into 4 levels. The accuracy of this training is 0.836.

```
> topfeatures(train.tokens.tfidf,n=10,decreasing = TRUE)
   energi     power     plant      link        rt  #nuclear      quot   #energi      year       via
1.9740879 1.7493841 1.6459363 1.5925422 1.4663765 1.4020809 1.3689921 0.9370970 0.7920722 0.7834900
> topfeatures(train.tokens.dfm, n=10, decreasing = TRUE)
 @mention   nuclear        rt      link     power    energi     plant  #nuclear      quot fukushima
      142       124        85        81        79        56        40        24        20         9
>
```

In general, the second training should have higher accuracy than the first one. But the

results came out wrong, so I compared the top words of DFM and TFIDF, it turned out that the TFIDF deleted some meaningless words. However, I can still not find the reason why the second accuracy is lower the first one.

The most important information I got from the emotion data is that the majority of people do not really know what it is since understanding nuclear energy needs the knowledge form different scientific fields. So the most content and sentiment about nuclear energy are neutral, only few can express the exact attitude (negative or positive) for this topic.