Assignment

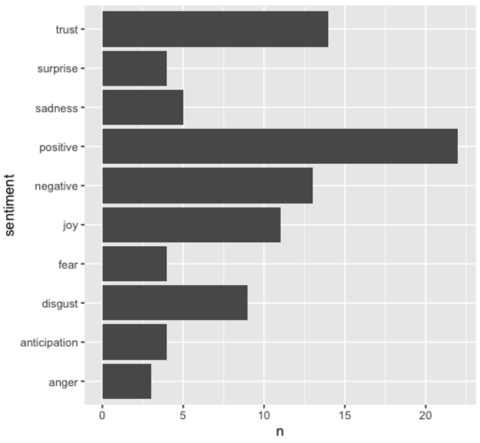
King Lear is a play authored by William Shakespeare. It is a classic work which portrays human emotions and foolish acts very well. The play is about a king named Lear and his three daughters who indulge in a clan war of sort, to inherit the legacy. As a result the kingdom is divided between the two daughters and the youngest one was exiled. After a while the king finds himself deserted by the two daughters who inherited the wealth and the kingdom. Later the exiled daughter returned with an army to take revenge on the siblings and the resulting battle culminated in the death of all four and more. This is an exemplary example of a tragic drama.

Sentiment analysis is a method of analysing the text corpus to glean insight into various emotions and feelings embedded. It classifies (a classification model) emotions broadly into three categories viz positive, negative or neutral. It includes techniques like Natural Language Processing, Artificial Intelligence to mine, extract and categorise the text.

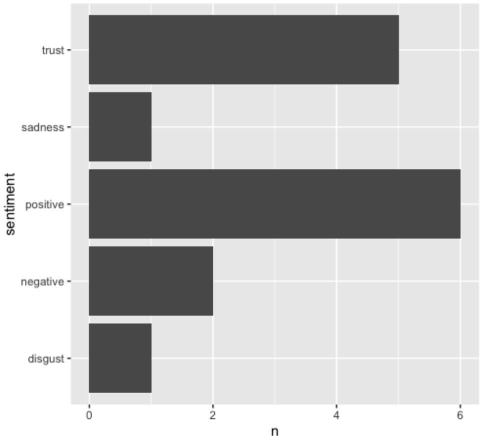
The assignment is about to do a sentiment analysis based on the dialogue between The King Lear and his youngest daughter, Cordelia.The scenes pertinent to their dialogue are found in the first and last part of the play. The model for analysis makes an implicit assumption that the consecutive conversation between the King and Cordelia forms the dialogue between them (we are not doing a deep semantic analysis). The distance between dialogues uttered by Cordelia is being tracked to distinguish between the first and the last part.

The system does csv file parsing (using command setwd() and read.csv()), then the canonicalization of data (using tidytext library). Preprocessing of the text was done by removing the stop words and custom stop words (using anti_join() in dplyr library). Calculated the “distance” (to distinguish it from metric) to segment the play into former and latter parts (of Cordelia and King Lear dialogues) . Extraction of the text based on the distance from the corpus (using custom logic based on R primitive commands). Tidytext library contains lexicons (aka dictionaries) like Affin, Bing and nrc. I have leveraged nrc for sentiment analysis on four segments (Lear words from first part and latter part , Cordelia words from first and latter part) to understand emotions of Lear and Cordelia in the first and the latter part. The resulting analysis was converted into a chart by leveraging the ggplot2 library from R. ggplot2 chart plotted sentiments against its count for Cordelia and Lear texts in the first and last part of the play accordingly.

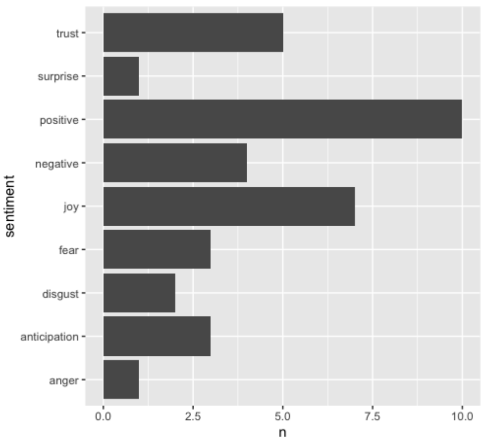
The analysis revealed that King Lear moved towards a positive frame of mind in the latter part of the drama. Whereas in the case of Cordelia sentiment was more on the negative side.



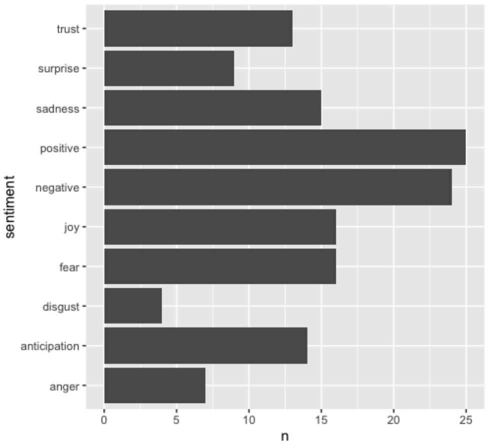
Cordelia first part



Cordelia last part



King Lear first part



King Lear last part

Appendix: R code

```
###################################################################
#  The assignment is about to do a sentiment analysis based on the dialogue
#  between The King Lear and his youngest daughter, Cordelia
#
#


install.packages("tidyverse") # data manipulation & plotting

library(dplyr) # manipulation of data
library(tidyr) # creating tidy data
library(stringr)  # manipulation of string
library(tidytext) # sentiment lexicon
library(data.table) # manipulating tabular data
library(ggplot2) # plot ggplot2 graph

##  set working directory and read the csv file
setwd("/Users/praseedpai/ACADEMIC/R Language/Assignment/Assignment")
King_Lear <- read.csv("King_Lear_words_and_players_only.csv")
str(King_Lear)

data(stop_words)  # a reference tibble of stop words in tidy format
nrow(stop_words)  # list of stop words

## unnest_tokens() converts to a tidy format anti_join() removes reference
stop words
King_Lear_tidy <- King_Lear%>%unnest_tokens(word,
text)%>%anti_join(stop_words)
# defining custom stop words
custom_stop_words <-
bind_rows(tibble(word=c("thou","thy","thee","tis"),lexicon=c("custom")),
stop_words)
# removal of custom stop words
King_Lear_tidy <- King_Lear_tidy%>%anti_join(custom_stop_words)
```

```
## variable to hold Lear first set of words from dialogue
## between Lear and Cordelia
King_Lear_words_act_first = c()
# variable to Cordelia hold first set of words from dialogue
## between Lear and Cordelia
Cordelia_words_act_first = c()
## variable to hold Lear last set of words from dialogue
## between Lear and Cordelia
King_Lear_words_act_last = c()
# variable to hold Cordelia last set of words from dialogue
## between Lear and Cordelia
Cordelia_words_act_last = c()
current_player=''
previous_player=''
next_player=''
# Flag to decide when the act to be split into two parts
act_break = FALSE
distance = 0
prev_item = 0
vector_distance =c()
vector_item = c()

## calculating the distance between two Cordelia
## so as to segment the play into two
## and the dialogue between Lear and Cordelia is in first
## last part of the play
for (item in 1:nrow(King_Lear_tidy)) {
  player = King_Lear_tidy[item,1]
  if (player == 'CORDELIA') {
    prev_distance = prev_item
    distance = item - prev_distance
    vector_distance= append(vector_distance,distance)
    vector_item = append(vector_item, item)
    prev_item = item
  }
}
df <- data.frame(vector_distance, vector_item)
## get the maximum distance
max_distance <- max(vector_distance)
## get the index of the maximum distance
```

```
df_distance <- df[ which(df$vector_distance==max_distance), ]

## Logic to extract the words by Cordelia first part and latter part
## also Lear words from first part and later part
for (item in 1:nrow(King_Lear_tidy)) {
  player = King_Lear_tidy[item,1]
  if (current_player == '') {
    current_player = King_Lear_tidy[item,1]
  } else if (current_player != player) {
    previous_player = current_player
    current_player = King_Lear_tidy[item,1]
  }
  if (current_player == 'CORDELIA' || current_player == 'LEAR') {
    flag = FALSE
    for (i in item:nrow(King_Lear_tidy)) {
      if (current_player != King_Lear_tidy[i,1]) {
        next_player = King_Lear_tidy[i,1]
        flag = TRUE
      }
      if (flag) {
        break;
      }
    }
  }
  # Breaks the play into two depending on the max distance
  if (length(df_distance$vector_item) == 1 && item >= df_distance$vector_item
&& act_break == FALSE) {
    act_break = TRUE
  }
  # collect words of Cordelia in the first part
  if (((current_player == 'CORDELIA' && previous_player == "LEAR") ||
       (current_player == 'CORDELIA' && next_player == 'LEAR')) &&
!act_break) {
    Cordelia_words_act_first =
append(Cordelia_words_act_first,King_Lear_tidy[item,2])
  }
  # collect words of Lear in the first part
  if (((current_player == "LEAR" && previous_player == "CORDELIA") ||
       (current_player == 'LEAR' && next_player == 'CORDELIA')) &&
!act_break) {
    King_Lear_words_act_first =
append(King_Lear_words_act_first,King_Lear_tidy[item,2])
  }
  # collect words of Cordelia in the last part
  if (((current_player == 'CORDELIA' && previous_player == "LEAR") ||
       (current_player == 'CORDELIA' && next_player == 'LEAR')) &&
act_break) {
    Cordelia_words_act_last =
append(Cordelia_words_act_last,King_Lear_tidy[item,2])
  }
  # enters words of Lear in the last part
  if (((current_player == "LEAR" && previous_player == "CORDELIA") ||
       (current_player == 'LEAR' && next_player == 'CORDELIA')) &&
act_break) {
    King_Lear_words_act_last =
append(King_Lear_words_act_last,King_Lear_tidy[item,2])
  }
}

nrc <- get_sentiments("nrc")
# nrc has 10 sentiment labels for word
table(nrc$sentiment)

# create data frame with player as Cordelia and word as cordelia first words
df_Cordelia_word_first <- data.frame(player ="CORDELIA", word =
Cordelia_words_act_first)
# join the cordelia words with nrc
cordelia_label_first <- df_Cordelia_word_first%>%inner_join(nrc)
table(cordelia_label_first$sentiment) # Cordelia sentiments in the first part
# group by the sentiment and find the count of each sentiment
cordelia_act_first <- cordelia_label_first%>%group_by(sentiment)%>%count()
# sentiments by player and word

# create data frame with player as Lear and word as Lear first words
df_Lear_word_first <- data.frame(player ="LEAR", word =
King_Lear_words_act_first)
# join the Lear words with nrc
lear_label_first <- df_Lear_word_first%>%inner_join(nrc)
table(lear_label_first$sentiment) # sentiments as table format
# group by the sentiment and find the count of each sentiment
```

```
lear_first_act <- lear_label_first%>%group_by(sentiment)%>%count()

# create data frame with player as Cordelia and word as cordelia last words
Cordelia_last_act_word <- data.frame(player ="CORDELIA", word =
Cordelia_words_act_last)
# join the cordelia words with nrc
cordelia_label_last_act <- Cordelia_last_act_word%>%inner_join(nrc)
table(cordelia_label_last_act$sentiment) # sentiments as table format
# group by the sentiment and find the count of each sentiment
cordelia_last_act <-
cordelia_label_last_act%>%group_by(sentiment)%>%count()

# create data frame with player as Lear and word as Lear Last words
Lear_word_last_act <- data.frame(player ="LEAR", word =
King_Lear_words_act_last)
# join the Lear words with nrc
lear_label_last_act<- Lear_word_last_act%>%inner_join(nrc)
table(lear_label_last_act$sentiment) # sentiments as table format
# group by the sentiment and find the count of each sentiment
lear_last_act <- lear_label_last_act%>%group_by(sentiment)%>%count()

## plotted graph of Lear words first part with sentiments against the count
lear_first_act %>%
  ggplot2::ggplot(aes(n, sentiment)) + ggplot2::geom_col()
## plotted graph of Lear words last part with sentiments against the count
lear_last_act %>%
  ggplot2::ggplot(aes(n, sentiment)) + ggplot2::geom_col()
## combined words of Lear words
ggplot() +
  geom_point(data =lear_first_act , aes(x = n, y = sentiment), color = "red") +
  geom_point(data = lear_last_act, aes(x = n, y = sentiment), color="green")

## plotted graph of Cordelia words first part with sentiments against the count
cordelia_act_first %>%
  ggplot2::ggplot(aes(n, sentiment)) + ggplot2::geom_col()
## plotted graph of Cordelia words first part with sentiments against the count
cordelia_last_act %>%
  ggplot2::ggplot(aes(n, sentiment)) + ggplot2::geom_col()
```

```
## combined graph of Lear words
ggplot() +
  geom_point(data = cordelia_act_first, aes(x = n, y = sentiment), color = "red")
+
  geom_point(data = cordelia_last_act, aes(x = n, y = sentiment), color =
"green")
```