Task1&2

Handing Zhang

11/29/2021

Task 1:

I picked ***The Game*** by Jack London

Task 2

Download Data and Explore

Download the Book **game** from gutenberg package.

```
game <- gutenberg_download(1160)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

```
# view(game)
```

Turn the dataset to a tidy form.

```
tidy_game <- game %>%
  unnest_tokens(word, text) %>%  # output is word column, input is from text column in original game da
  anti_join(stop_words) # get rid of stop words
```

```
## Joining, by = "word"
```

```
tidy_game <- game %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,       # add a chapter column to mark chapter number.
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE))))%>%
   unnest_tokens(word, text) %>%
   anti_join(stop_words) # get rid of stop words.
```

```
## Joining, by = "word"
```

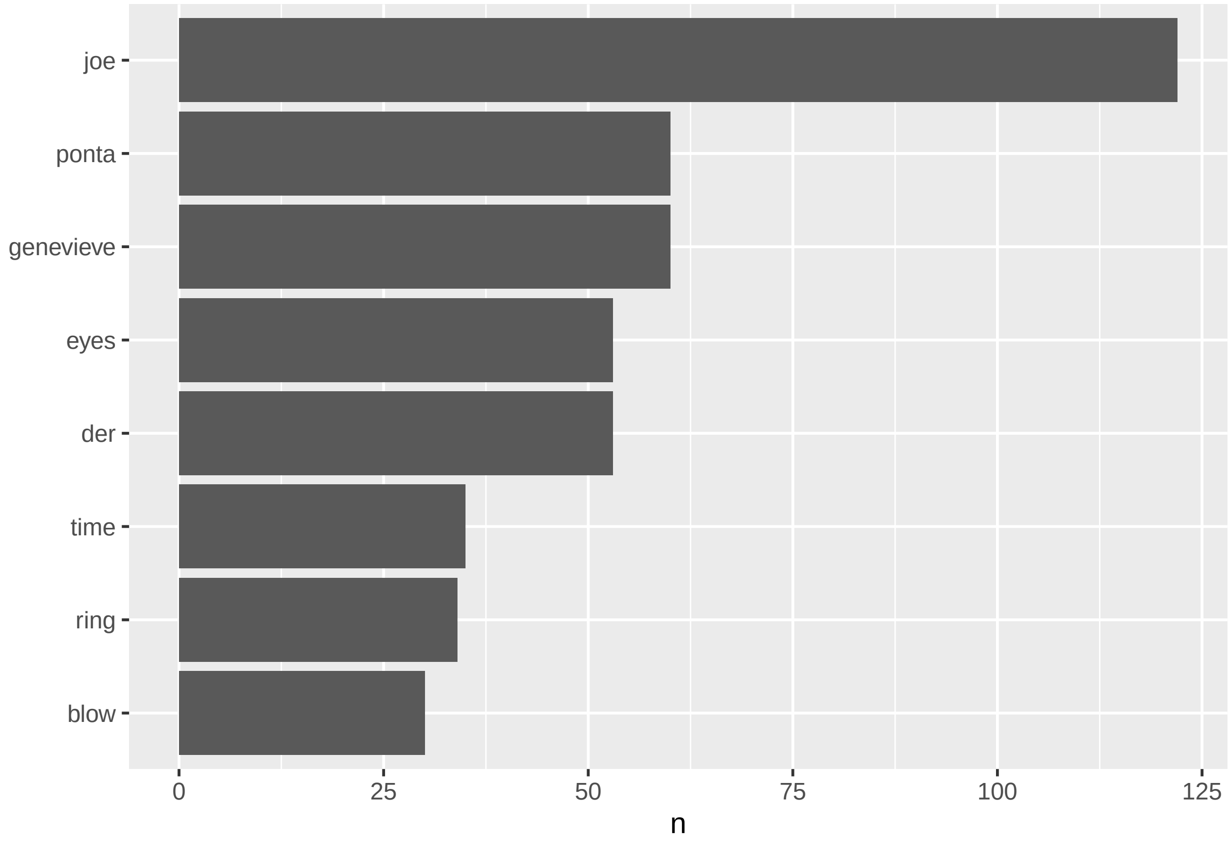
We start by looking at the most frequently appeared words in the book.

```
tidy_game %>%
  count(word, sort = T)
```

```
## # A tibble: 2,486 x 2
##    word            n
##    <chr>       <int>
##  1 joe           122
##  2 genevieve      60
##  3 ponta          60
##  4 der            53
##  5 eyes           53
##  6 time           35
##  7 ring           34
##  8 blow           30
##  9 ponta’s        26
## 10 silverstein    26
## # ... with 2,476 more rows
```

let’s also visualize the words that appeared more than 30 times in a descending order.

```
tidy_game %>%
  count(word, sort = T) %>%
  filter(n >= 30) %>%
  ggplot(aes(x = n, y = reorder(word, n))) +
  geom_col() +
  labs(y = NULL)
```



Let’s calculate the frequency of each word

```
frequency <- tidy_game %>%
  mutate(word = str_extract(word, "[a-z’]+")) %>%
  ## eliminate underscores around words so that _apple_ is treated thesame as apple.
  count(word) %>%
  mutate(proportion = n / sum(n)) %>%
  arrange(desc(proportion))

frequency
```

```
## # A tibble: 2,486 x 3
##    word            n proportion
##    <chr>       <int>      <dbl>
##  1 joe           122    0.0220
##  2 genevieve      60    0.0108
##  3 ponta          60    0.0108
##  4 der            53    0.00955
##  5 eyes           53    0.00955
##  6 time           35    0.00630
##  7 ring           34    0.00612
##  8 blow           30    0.00540
##  9 ponta’s        26    0.00468
## 10 silverstein    26    0.00468
## # ... with 2,476 more rows
```

Sentimental Analysis

Get sentiment words from sentiment lexicons “AFINN” “BING” “NRC”

```
afinn <- get_sentiments("afinn")
bing <- get_sentiments("bing")
```

```
# textdata::lexicon_nrc(delete = TRUE)
# nrc <- textdata::lexicon_nrc()
# write.csv(nrc, "/Users/handingzhang/Desktop/mssp/MA 615/Homework/615-Assignment-4/nrc.csv", row.names
nrc <- read.csv("nrc.csv")
```

Now let’s see the most frequently used word with “joy” sentiment according to nrc in **The game**

```
nrc_joy <- nrc %>%
  filter(sentiment == "joy")
# nrc_joy
# we take out all words with joy sentiment from nrc.


# use inner_join to join the rows of tidy_game that has the according elements
tidy_game %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
## # A tibble: 115 x 2
##    word          n
##    <chr>     <int>
##  1 love         19
##  2 clean        12
##  3 beautiful     9
##  4 beauty        9
##  5 found         8
##  6 lover         8
##  7 cream         7
##  8 money         7
##  9 delight       6
## 10 embrace       5
## # ... with 105 more rows
```

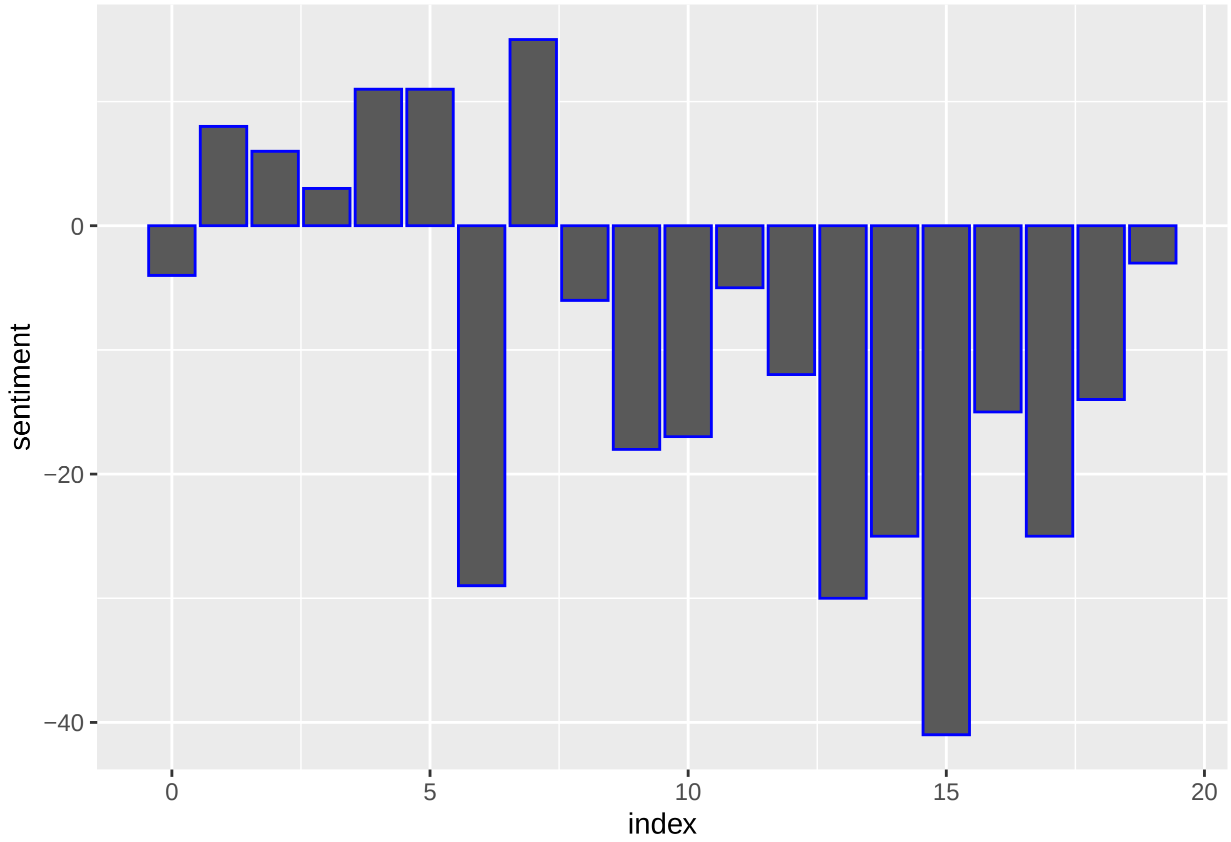
We get a sentiment score for each 80 lines by the number of positive and negative sentimental words according to nrc.

```
game_sentiment <- tidy_game %>%
  inner_join(bing) %>%
  count(index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

Plot the sentiment score by nrc measure against timeline of the book by index of 80 lines.

```
ggplot(game_sentiment, aes(index, sentiment)) +
  geom_col(show.legend = FALSE, color = "blue")
```



We see in general the sentiment is quite negative, but we also notice that at one point the sentinent is fairly high.

```
which(game_sentiment$sentiment >= 10)
```

```
## [1] 5 6 8
```

There might be a positive plot happening between line 102 * 80 = 8160 and 103 * 80 = 8240.

Now let's compare the three lexicons.

```
# Measured by afinn
game_afinn <- tidy_game %>%
  inner_join(afinn) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")
```

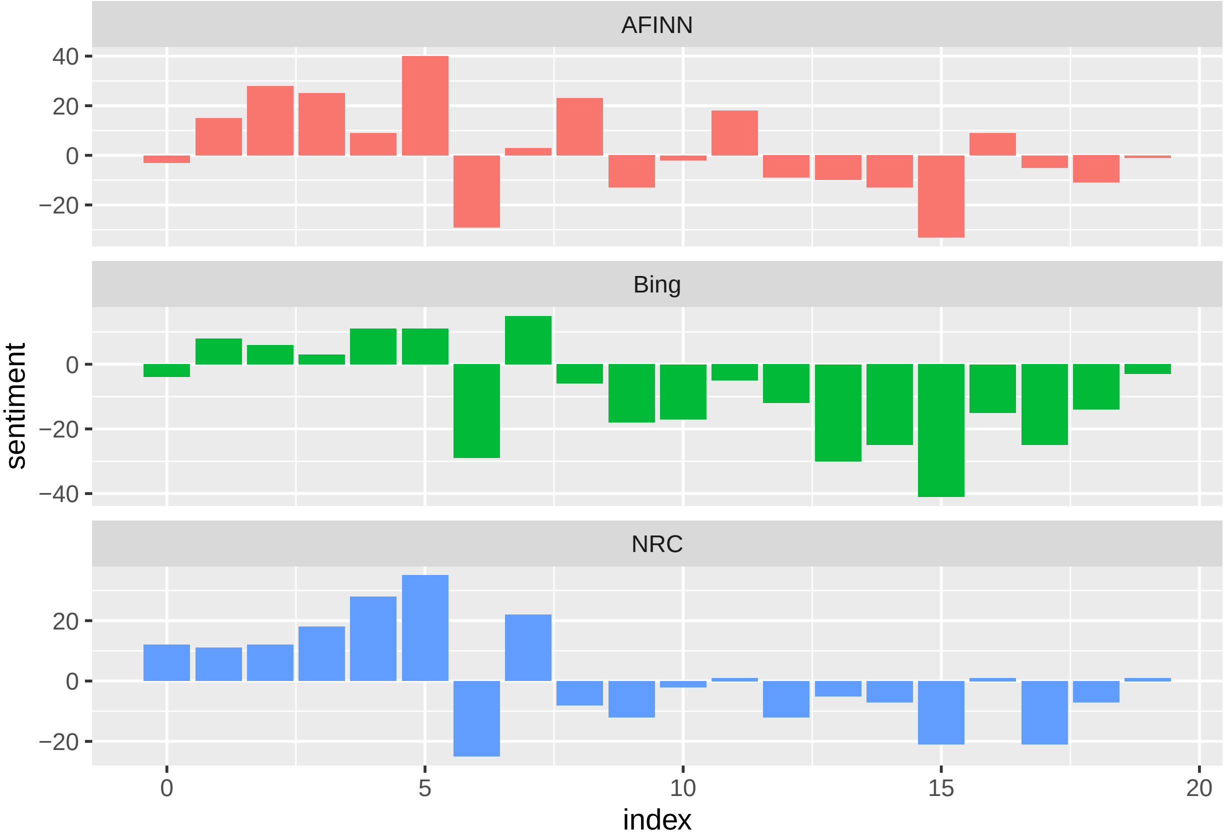
```
## Joining, by = "word"
```

```
# Measured by bing and nrc
game_bing_and_nrc <- bind_rows(
  tidy_game %>%
    inner_join(bing) %>%
    mutate(method = "Bing"),
  tidy_game %>%
    inner_join(nrc %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))) %>%
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

Compare the visualization of sentiment measurements by the three methods

```
bind_rows(game_afinn,
          game_bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



Count the number of each word in each sentiment for being.

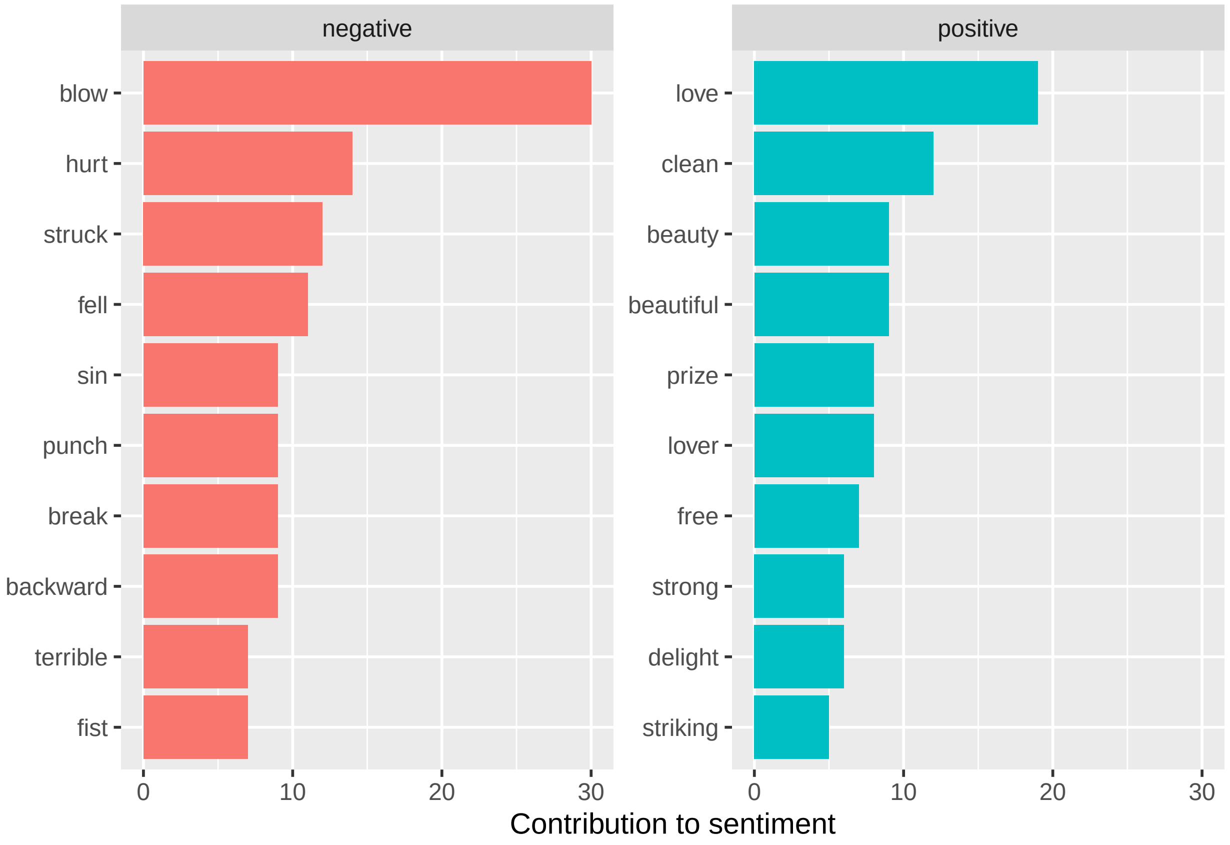
```
game_bing_word_counts <- tidy_game %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
game_bing_word_counts
```

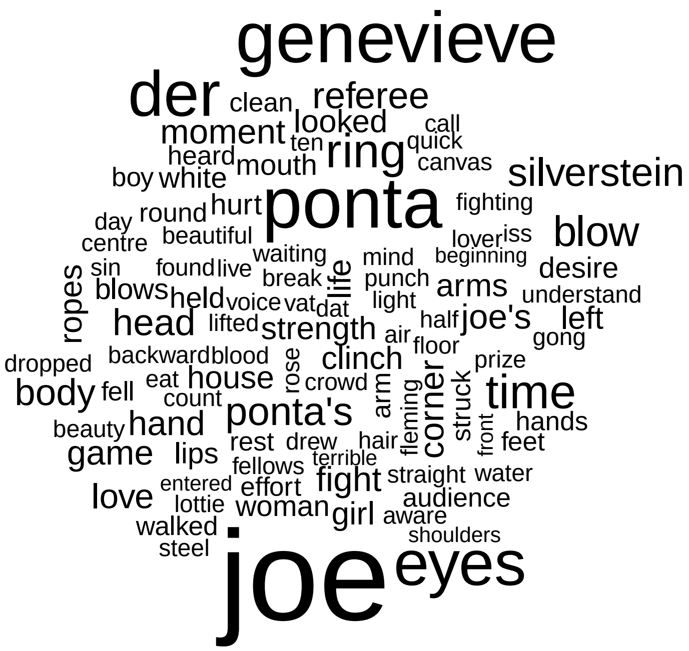
```
## # A tibble: 541 x 3
##    word      sentiment     n
##    <chr>     <chr>     <int>
##  1 blow      negative     30
##  2 love      positive     19
##  3 hurt      negative     14
##  4 clean     positive     12
##  5 struck    negative     12
##  6 fell      negative     11
##  7 backward  negative      9
##  8 beautiful positive      9
##  9 beauty    positive      9
## 10 break     negative      9
## # ... with 531 more rows
```

```
game_bing_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
```



Make a word cloud

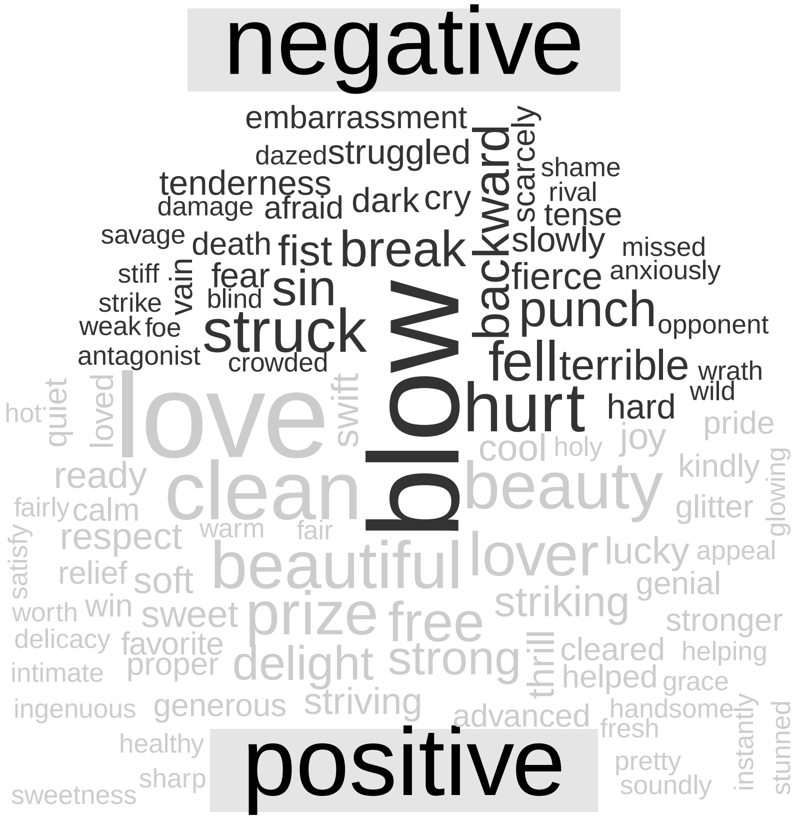
```r
tidy_game %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



Word cloud with positive sentiments blow and neggative above.

```
tidy_game %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                   max.words = 100)
```

```
## Joining, by = "word"
```


negative
embarrassment
dazed
struggled
backward
scarcely
shame
rival
tense
tenderness
damage
afraid
dark
cry
savage
death
fist
break
slowly
missed
stiff
vain
fear
sin
fierce
anxiously
strike
blind
punch
opponent
weak
foe
struck
blow
antagonist
crowded
fell
terrible
wrath
wild
hurt
hard
hot
quiet
loved
love
swift
pride
cool
holy
joy
ready
clean
beauty
kindly
glowing
fairly
calm
glitter
satisfy
respect
warm
fair
beautiful
lover
lucky
appeal
relief
soft
genial
worth
win
sweet
prize
free
striking
stronger
delicacy
favorite
thrill
cleared
helping
intimate
proper
delight
strong
helped
grace
ingenuous
generous
striving
advanced
handsome
fresh
instantly
stunned
healthy
positive
pretty
sharp
soundly
sweetness