Harry Potter Sentiment Analysis

Gabriel Benitez

23 de diciembre de 2017

Importing the libraries neccesary to create the sentimental Analysis

Import all the books and gather them in one dataframe. First with rbind I will put them all thogehter. Each book is a character so first I create the column book to separate it. Then I create key and value for the texts so the chapter can be the key and the text wil be separated in words with unnest_tokens function

```
hp_words <- list(philosophers_stone = philosophers_stone,
                 chamber_of_secrets = chamber_of_secrets,
                 prisoner_of_azkaban = prisoner_of_azkaban,
                 goblet_of_fire = goblet_of_fire,
                 order_of_the_phoenix = order_of_the_phoenix,
                 half_blood_prince = half_blood_prince,
                 deathly_hallows = deathly_hallows
                 )%>%
              ldply(rbind)%>%
              mutate(book = factor(seq_along(.id),labels = .id))%>%
              select(-.id)%>%
              gather(key= "chapter",value="text", -book)%>%
              filter(!is.na(text))%>%
              mutate(chapter=as.integer(chapter))%>%
              unnest_tokens(word,text,token="words")
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

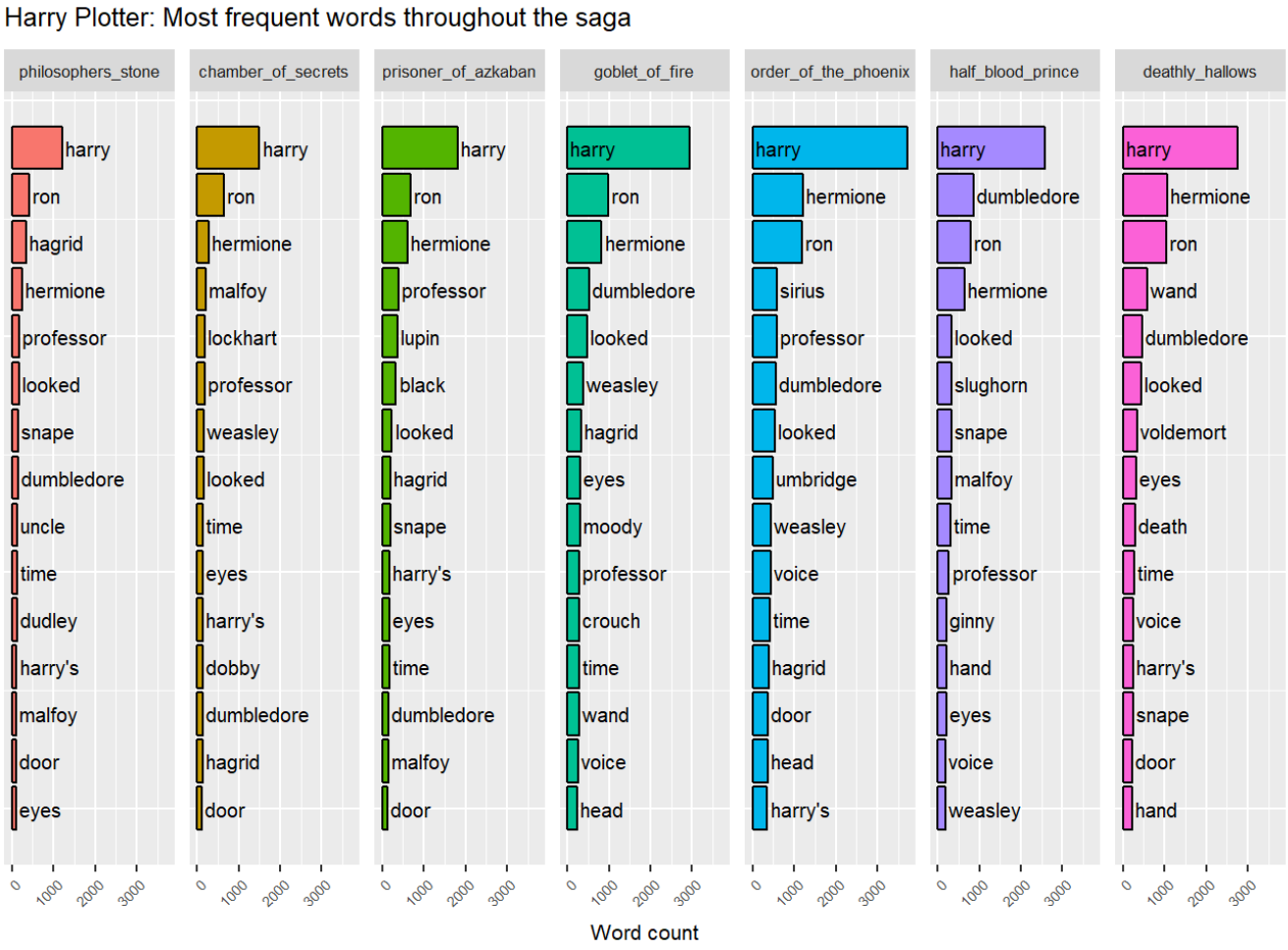
```
head(hp_words)
```

```
##                    book chapter  word
## 1   philosophers_stone       1   the
## 1.1 philosophers_stone       1   boy
## 1.2 philosophers_stone       1   who
## 1.3 philosophers_stone       1 lived
## 1.4 philosophers_stone       1    mr
## 1.5 philosophers_stone       1   and
```

Now lets see the most frecuent words per book. First I used group by to put together words and book. After that with anti_join I eliminate all the stop_words just like “of”,“from” etc. Wit seq_along I order them and with filter I left only the 15 more common words.

Ggplot2 is a good tool to create beutiful graphs, geom_bar is the funcionto create bar charts, in this case I tell it to use y as the count of words and x with the word

```
hp_words%>%
  group_by(book,word)%>%
  anti_join(stop_words,by = "word")%>%
  count()%>%
  arrange(desc(n))%>%
  group_by(book)%>%
  mutate(top= seq_along(word))%>%
  filter(top<=15)%>%
  ggplot(aes(x = -top,fill = book)) +
  geom_bar(aes(y = n), stat = 'identity', col = 'black') +
  # make sure words are printed either in or next to bar
  geom_text(aes(y = ifelse(n > max(n) / 2, max(n) / 50, n + max(n) / 50),
                label = word), size = 8/3, hjust = "left") +
  theme(legend.position = 'none',
        text = element_text(size =8 ),
        axis.text.x = element_text(angle = 45, hjust = 1, size = 8/1.5), # rotate
x text
        axis.ticks.y = element_blank(), # remove y ticks
        axis.text.y = element_blank()) + # remove y text
  labs(y = "Word count", x = "", # add labels
       title = "Harry Plotter: Most frequent words throughout the saga") +
  facet_grid(. ~ book)+  # separate plot for each book
coord_flip() # flip axes
```

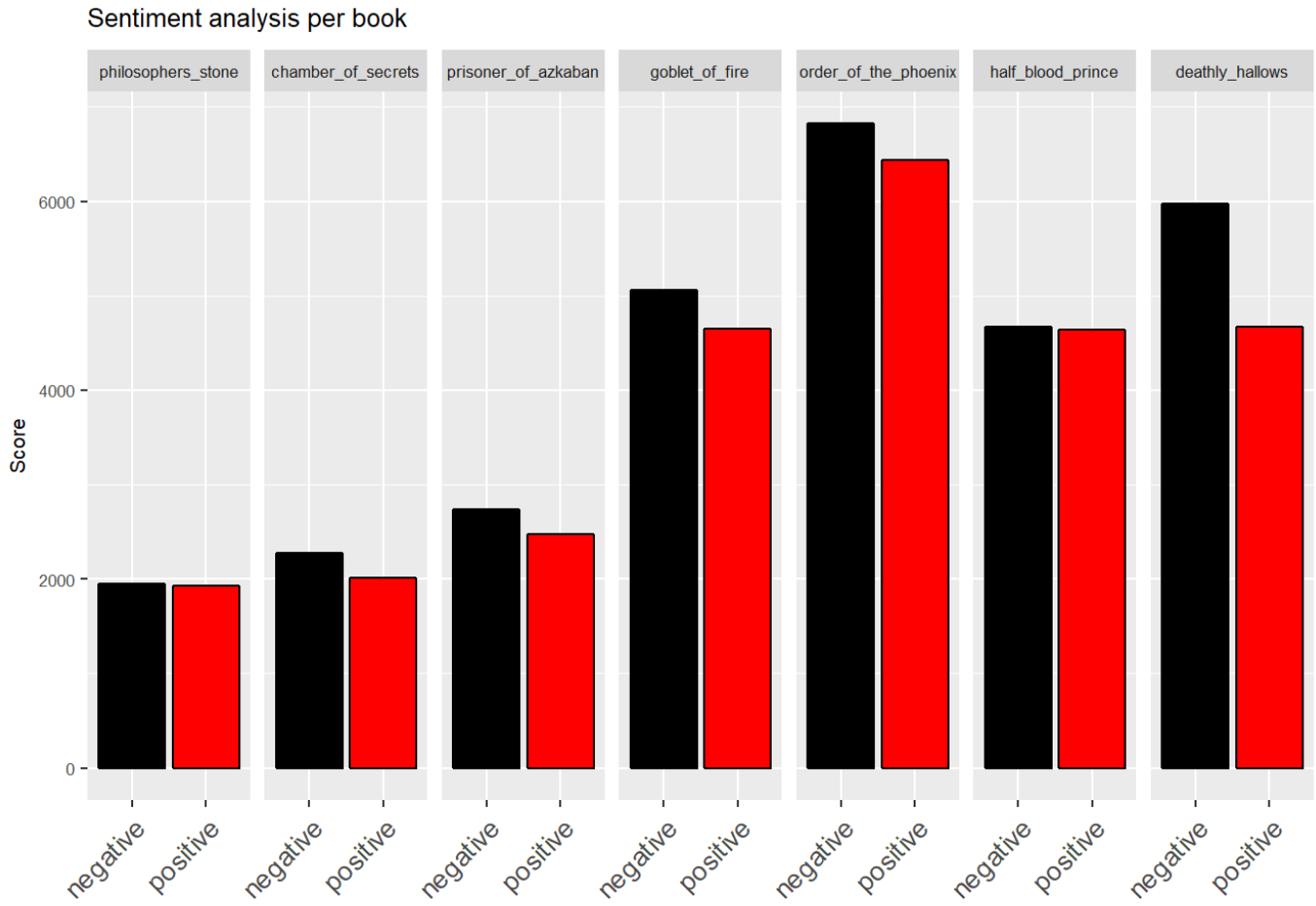


Ofc Harry is the most important word in the book , but its interesting too see the importance of words like dumbledore increases over the books. Also is interesting that the ord death is releveant only until the last book.

Sentiment Analysis per book

Now that We have seen the most frecuent words per book is time to analyze each

```
hp_senti <- bind_rows(
  # 1 AFINN
  hp_words %>%
    inner_join(get_sentiments("afinn"), by = "word") %>%
    filter(score != 0) %>% # delete neutral words
    mutate(sentiment = ifelse(score < 0, 'negative', 'positive')) %>% # identify s
entiment
    mutate(score = sqrt(score ^ 2)) %>% # all scores to positive
    group_by(book, chapter, sentiment) %>%
    mutate(dictionary = 'afinn'))
hp_senti%>%
  group_by(book,sentiment)%>%
  count()%>%
  ggplot(aes(y=n, x=sentiment, fill = sentiment)) +
  geom_bar( stat = 'identity', col ='black') +
    theme(legend.position = 'none',
        text = element_text(size =8 ),
        axis.text.x = element_text(angle = 45, hjust = 1, size = 10) # rotate x te
xt
         # remove y ticks
        ) + # remove y text
  labs(y = "Score", x = "", # add labels
       title = "Sentiment analysis per book") +
  facet_grid(. ~ book) + scale_fill_manual(values= c("#000000","#FF0000"))
```



Its interesting that the negative score is always higher than the positive score. It also helps to look for the “darkest” book which is the last one and the fith one.