Individual\_assignment\_movie\_spider.R

Huangyu3

2020-02-11

library(magrittr)

## Warning: package 'magrittr' was built under R version 3.6.2

library(rvest)

## Warning: package 'rvest' was built under R version 3.6.2

## Loading required package: xml2

## Warning: package 'xml2' was built under R version 3.6.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.6.2

## -- Attaching packages ------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v stringr 1.4.0  
## v tidyr 1.0.0 v forcats 0.4.0  
## v readr 1.3.1

## Warning: package 'forcats' was built under R version 3.6.2

## -- Conflicts --------------------------------------------- tidyverse\_conflicts() --  
## x tidyr::extract() masks magrittr::extract()  
## x dplyr::filter() masks stats::filter()  
## x readr::guess\_encoding() masks rvest::guess\_encoding()  
## x dplyr::lag() masks stats::lag()  
## x purrr::pluck() masks rvest::pluck()  
## x purrr::set\_names() masks magrittr::set\_names()

library(tidytext)

## Warning: package 'tidytext' was built under R version 3.6.2

library(stringr)  
library(ggplot2)  
library(reshape2)

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.6.2

## Loading required package: RColorBrewer

library(tidyverse)  
library(tidyr)  
  
  
  
Joker <- xml2::read\_html("https://www.imdb.com/title/tt7286456/reviews?ref\_=tt\_urv")   
Joker\_review <- Joker %>%  
 html\_nodes('.text') %>%  
 html\_text()  
#View(Joker\_review)   
  
The\_Dark\_Knight <-xml2::read\_html("https://www.imdb.com/title/tt0468569/reviews?ref\_=tt\_urv")  
The\_Dark\_Knight\_review <- The\_Dark\_Knight %>%  
 html\_nodes('.text') %>%  
 html\_text()  
#View(The\_Dark\_Knight\_review)  
  
df\_Joker <- data\_frame(id=1:25, text=Joker\_review)

## Warning: `data\_frame()` is deprecated, use `tibble()`.  
## This warning is displayed once per session.

#View(df\_Joker)  
  
df\_The\_Dark\_Knight <- data\_frame(id=1:25, text= The\_Dark\_Knight\_review)  
#View(df\_The\_Dark\_Knight)  
  
# cust\_stop <- data\_frame(  
# word=c("movie","film","movies"),  
# lexicon=rep("custom",each=3)   
# )  
  
# data(stop\_words)  
# afinn <- get\_sentiments("afinn")  
# nrc <- get\_sentiments("nrc")  
# bing <- get\_sentiments("bing")  
  
bigrams\_separated\_Joker <- df\_Joker %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n=2)%>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_separated\_Knight <- df\_The\_Dark\_Knight %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n=2)%>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigram\_counts\_Joker <- bigrams\_separated\_Joker %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word) %>%  
 count(word1, word2, sort = TRUE)  
  
bigram\_counts\_Knight <- bigrams\_separated\_Knight %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word) %>%  
 count(word1, word2, sort = TRUE)  
  
# Joker\_review\_frequencies <- df\_Joker %>%  
# unnest\_tokens(bigram, text, token = "ngrams", n=2)%>%  
# separate(bigram, c("word1", "word2"), sep = " ") %>%  
# filter(!word1 %in% stop\_words$word) %>%  
# filter(!word2 %in% stop\_words$word) %>%  
# unite(bigram, word1, word2, sep=" ") %>%  
# count(id, bigram) %>%  
# bind\_tf\_idf(bigram, id, n) %>%  
# arrange(desc(tf\_idf))  
   
# The\_Dark\_Knight\_review\_frequencies <- df\_The\_Dark\_Knight %>%  
# unnest\_tokens(bigram, text, token = "ngrams", n=2)%>%  
# separate(bigram, c("word1", "word2"), sep = " ") %>%  
# filter(!word1 %in% stop\_words$word) %>%  
# filter(!word2 %in% stop\_words$word) %>%  
# unite(bigram, word1, word2, sep=" ") %>%  
# count(id, bigram) %>%  
# bind\_tf\_idf(bigram, id, n) %>%  
# arrange(desc(tf\_idf))   
  
negation\_tokens <- c("no","never","without","not")  
  
negated\_words\_Joker <- bigrams\_separated\_Joker %>%  
 filter(word1 %in% negation\_tokens) %>%  
 inner\_join(get\_sentiments("afinn"), by=c(word2="word")) %>%  
 count(word1, word2, value, sort=TRUE) %>%  
 ungroup()  
  
negated\_words\_Knight <- bigrams\_separated\_Knight %>%  
 filter(word1 %in% negation\_tokens) %>%  
 inner\_join(get\_sentiments("afinn"), by=c(word2="word")) %>%  
 count(word1, word2, value, sort=TRUE) %>%  
 ungroup()  
  
library(igraph)

## Warning: package 'igraph' was built under R version 3.6.2

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:purrr':  
##   
## compose, simplify

## The following object is masked from 'package:tidyr':  
##   
## crossing

## The following object is masked from 'package:tibble':  
##   
## as\_data\_frame

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

#install.packages("ggraph")  
library(ggraph)

## Warning: package 'ggraph' was built under R version 3.6.2

bigram\_graph <- bind\_rows(bigrams\_separated\_Knight,  
 bigrams\_separated\_Joker)%>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word) %>%  
 count(word1, word2, sort = TRUE) %>%  
 filter(n>1) %>%  
 graph\_from\_data\_frame ()  
  
bigram\_graph

## IGRAPH 8ce55d4 DN-- 134 100 --   
## + attr: name (v/c), n (e/n)  
## + edges from 8ce55d4 (vertex names):  
## [1] dark ->knight heath ->ledger comic ->book   
## [4] batman ->begins christopher->nolan harvey ->dent   
## [7] bruce ->wayne christian ->bale joaquin ->phoenix   
## [10] aaron ->eckhart gotham ->city batman ->movie   
## [13] heath ->ledger's ledger's ->performance 10 ->10   
## [16] ledger's ->joker michael ->caine morgan ->freeman   
## [19] attorney ->harvey book ->movie district ->attorney   
## [22] gary ->oldman jack ->nicholson superhero ->movie   
## + ... omitted several edges

ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link()+  
 geom\_node\_point()+  
 geom\_node\_text(aes(label=name), vjust =1, hjust=1)

