Analise de Redes Sociais e Text Mining - Trabalho 4

26/Setembro/2018

A56843646 Ricardo Squassina Lee, A56089622 Carlos Augusto Polato, A56865193 Gustavo Hotta, A56871487 Rafael Furlan

## Realizar uma análise de sentimento. Para a análise de sentimento criar:

1. word cloud,
2. word cloud positiva,
3. word cloud negativa,
4. semantic network,
5. word cloud bigram

rm(list = ls())  
cat("\014")

if(!require(tidytext)) {  
 install.packages("tidytext")  
 library(tidytext)  
}  
if(!require(dplyr)) {  
 install.packages("dplyr")  
 library(dplyr)  
}  
if(!require(stringr)) {  
 install.packages("stringr")  
 library(stringr)  
}  
if(!require(tm)) {  
 install.packages("tm")  
 library(tm)  
}  
if(!require(tidyr)) {  
 install.packages("tidyr")  
 library(tidyr)  
}  
if(!require(gutenbergr)) {  
 install.packages("gutenbergr")  
 library(gutenbergr)  
}  
if(!require(scales)) {  
 install.packages("scales")  
 library(scales)  
}  
if(!require(ggplot2)) {  
 install.packages("ggplot2")  
 library(ggplot2)  
}  
if(!require(wordcloud)) {  
 install.packages("wordcloud")  
 library(wordcloud)  
}  
if(!require(reshape2)) {  
 install.packages("reshape2")  
 library(reshape2)  
}  
if(!require(igraph)) {  
 install.packages("igraph")  
 library(igraph)  
}  
if(!require(ggraph)) {  
 install.packages("ggraph")  
 library(ggraph)  
}  
if(!require(widyr)) {  
 install.packages("widyr")  
 library(widyr)  
}

## Leitura do Texto

Para o trabalho foram escohidos textos relativos à Segunda Guerra Mundial.

France <- gutenberg\_download(c(17813, 16437, 45542, 18483,9975), meta\_fields = "title")

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

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

Dardanelles <- gutenberg\_download(c(11008, 11513, 15896, 3306, 35119), meta\_fields = "title")  
#Russia <- gutenberg\_download(c(46191, 10967, 22523, 10972, 53482), meta\_fields = "title")  
Russia <- gutenberg\_download(c(46191, 10967, 22523, 10972), meta\_fields = "title")  
  
  
france\_books <- France %>%  
 group\_by(gutenberg\_id) %>%  
 mutate(linenumber = row\_number()) %>%  
 ungroup()  
  
dardanelles\_books <- Dardanelles %>%  
 group\_by(gutenberg\_id) %>%  
 mutate(linenumber = row\_number()) %>%  
 ungroup()  
  
russia\_books <- Russia %>%  
 group\_by(gutenberg\_id) %>%  
 mutate(linenumber = row\_number()) %>%  
 ungroup()

## Separando as palavras contidas nos textos que estão sendo analisados e removendo as “stop words”

france\_tidy <- france\_books %>%   
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 drop\_na()  
  
dardanelles\_tidy <- dardanelles\_books %>%   
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 drop\_na()  
  
russia\_tidy <- russia\_books %>%   
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 drop\_na()

## Contando as palavras para uma primeira análise da frequência:

count(france\_tidy, word, sort=TRUE) %>% top\_n(10)

## # A tibble: 10 x 2  
## word n  
## <chr> <int>  
## 1 french 599  
## 2 war 589  
## 3 day 518  
## 4 german 441  
## 5 time 422  
## 6 paris 396  
## 7 france 386  
## 8 captain 248  
## 9 germans 237  
## 10 left 222

count(dardanelles\_tidy, word, sort=TRUE) %>% top\_n(10)

## # A tibble: 10 x 2  
## word n  
## <chr> <int>  
## 1 ken 699  
## 2 time 444  
## 3 roy 373  
## 4 day 312  
## 5 night 294  
## 6 war 259  
## 7 left 246  
## 8 water 244  
## 9 turks 239  
## 10 front 217

count(russia\_tidy, word, sort=TRUE) %>% top\_n(10)

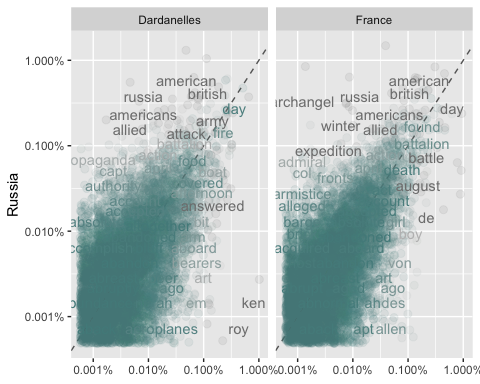
## # A tibble: 10 x 2  
## word n  
## <chr> <int>  
## 1 russian 965  
## 2 american 798  
## 3 british 568  
## 4 russia 520  
## 5 company 480  
## 6 archangel 457  
## 7 time 434  
## 8 front 415  
## 9 soldiers 383  
## 10 day 382

## Analisando a frequência das palavras nos conjuntos de texto

# Frequency as per the book  
#frequency <- bind\_rows(mutate(france\_tidy, subject = "France"),   
# mutate(dardanelles\_tidy, subject = "Dardanelles"),  
# mutate(russia\_tidy, subject = "Russia")) %>%   
# mutate(word = str\_extract(word, "[a-z']+")) %>%  
# count(subject, word) %>%  
# group\_by(subject) %>%  
# mutate(proportion = n / sum(n)) %>%   
# select(-n) %>%   
# spread(subject, proportion) %>%   
# gather(subject, proportion,`Personal Narratives`:`WWII Fiction`)  
  
# Frequency step-by-step  
binded\_texts <- bind\_rows(mutate(france\_tidy, subject = "France"),   
 mutate(dardanelles\_tidy, subject = "Dardanelles"),  
 mutate(russia\_tidy, subject = "Russia"))  
  
counted\_texts <- count(binded\_texts, subject, word)  
  
grouped\_texts <- group\_by(counted\_texts,subject)  
  
grouped\_texts\_by\_proportion <- mutate(grouped\_texts, proportion = n / sum(n))  
  
selected\_texts <- select(grouped\_texts\_by\_proportion,-n)  
  
spreaded\_text <- spread(selected\_texts, subject, proportion)  
  
frequency <- gather(spreaded\_text,subject, proportion,`Dardanelles`:`France`)

## Gráfico de proporção das palavras dos livros sobre a França e Dardanelles vs. Russia:

ggplot(frequency, aes(x = proportion, y = `Russia`, color = abs(`Russia` - proportion))) +  
 geom\_abline(color = "gray40", lty = 2) +  
 geom\_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +  
 geom\_text(aes(label = word), check\_overlap = TRUE, vjust = 1.5) +  
 scale\_x\_log10(labels = percent\_format()) +  
 scale\_y\_log10(labels = percent\_format()) +  
 scale\_color\_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray75") +  
 facet\_wrap(~subject, ncol = 2) +  
 theme(legend.position="none") +  
 labs(y = "Russia", x = NULL)



## Teste de Correlação entre os textos

cor.test(data = frequency[frequency$subject == "Dardanelles",],  
 ~ proportion + `Russia`)

##   
## Pearson's product-moment correlation  
##   
## data: proportion and Russia  
## t = 51.042, df = 7470, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.4915007 0.5251281  
## sample estimates:  
## cor   
## 0.5085082

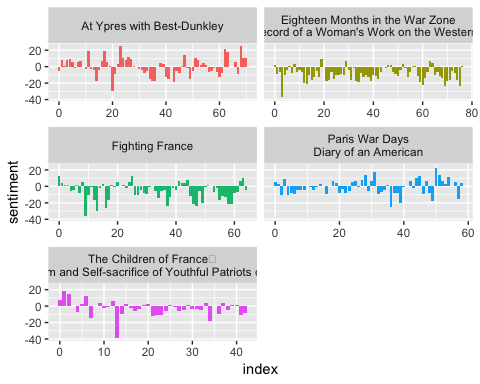
cor.test(data = frequency[frequency$subject == "France",],  
 ~ proportion + `Russia`)

##   
## Pearson's product-moment correlation  
##   
## data: proportion and Russia  
## t = 65.76, df = 7805, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5826256 0.6111792  
## sample estimates:  
## cor   
## 0.5970915

## Análise de Sentimentos

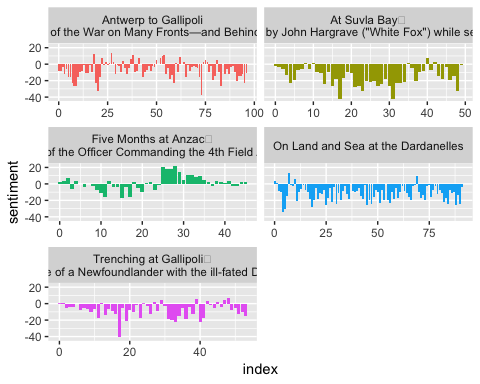
### Análise de sentimentos dos livros da França:

france\_sentiment <- france\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(title, index = linenumber %/% 80, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)  
  
ggplot(france\_sentiment, aes(index, sentiment, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_x")



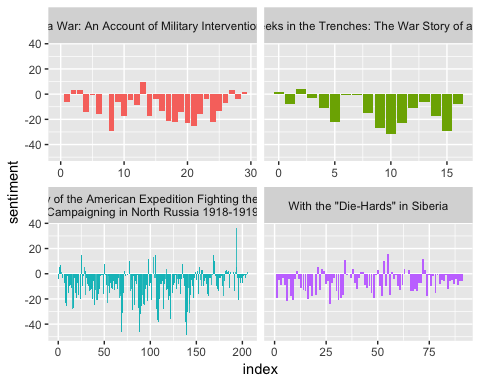
### Análise de sentimentos dos livros de Dardonelles:

dardanelles\_sentiment <- dardanelles\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(title, index = linenumber %/% 80, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)  
  
ggplot(dardanelles\_sentiment, aes(index, sentiment, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_x")



### Análise de sentimentos dos livros da Rússia:

russia\_sentiment <- russia\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(title, index = linenumber %/% 80, sentiment) %>%  
 spread(sentiment, n, fill = 0) %>%  
 mutate(sentiment = positive - negative)  
  
ggplot(russia\_sentiment, aes(index, sentiment, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_x")

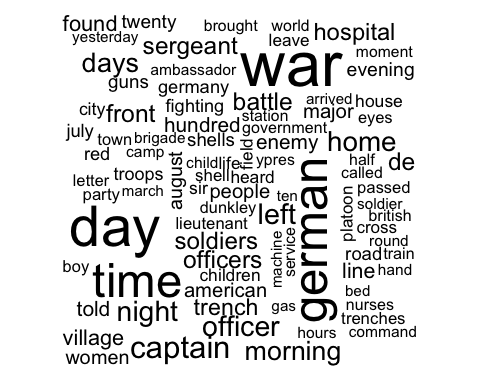


Notamos, pelos gráficos, que são histórias tristes, confirmando o que se espera de livros de histórias reais da Segunda Guerra Mundial.

## Nuvens de Palavras

### Nuvem de palavras dos livros da França:

france\_tidy %>%  
 anti\_join(stop\_words) %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 100))



### Nuvem de palavras dos livros de Dardanelles:

dardanelles\_tidy %>%  
 anti\_join(stop\_words) %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 100))



### Nuvem de palavras dos livros da Rússia:

russia\_tidy %>%  
 anti\_join(stop\_words) %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 100))



### Nuvem de palavras com divisão dos sentimentos dos livros da França:

france\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(colors = c("gray20", "gray80"),  
 max.words = 50)



### Nuvem de palavras com divisão dos sentimentos dos livros de Dardanelles:

dardanelles\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(colors = c("gray20", "gray80"),  
 max.words = 50)



### Nuvem de palavras com divisão dos sentimentos dos livros da Rússia:

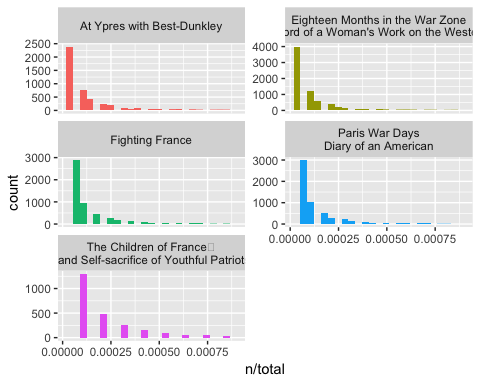
russia\_tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(colors = c("gray20", "gray80"),  
 max.words = 50)



## Análise da frquência de palavras nos diferentes livros:

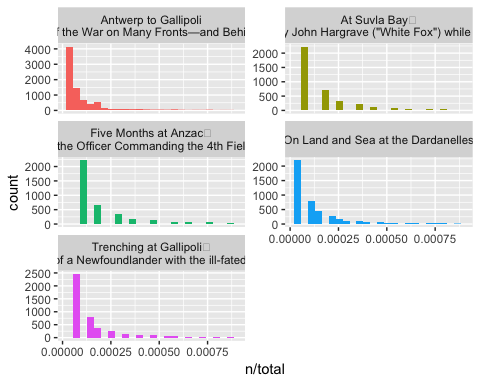
### França:

france\_book\_words <- france\_books %>%  
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 count(title, word, sort = TRUE) %>%  
 ungroup()  
  
france\_total\_words <- france\_book\_words %>%   
 group\_by(title) %>%   
 summarize(total = sum(n))  
  
france\_book\_words <- left\_join(france\_book\_words, france\_total\_words)  
  
ggplot(france\_book\_words, aes(n/total, fill = title)) +  
 geom\_histogram(show.legend = FALSE, bins = 25) +  
 xlim(NA, 0.0009) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_y")



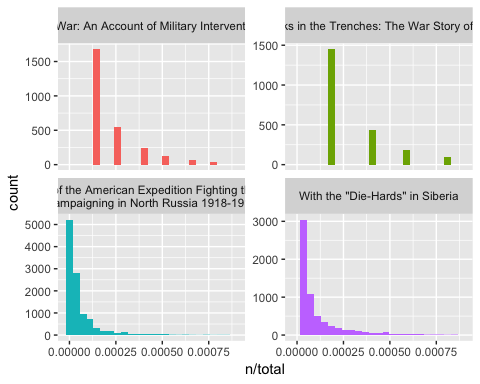
### Dardanelles:

dardanelles\_book\_words <- dardanelles\_books %>%  
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 count(title, word, sort = TRUE) %>%  
 ungroup()  
  
dardanelles\_total\_words <- dardanelles\_book\_words %>%   
 group\_by(title) %>%   
 summarize(total = sum(n))  
  
dardanelles\_book\_words <- left\_join(dardanelles\_book\_words, dardanelles\_total\_words)  
  
ggplot(dardanelles\_book\_words, aes(n/total, fill = title)) +  
 geom\_histogram(show.legend = FALSE, bins = 25) +  
 xlim(NA, 0.0009) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_y")



### Rússia

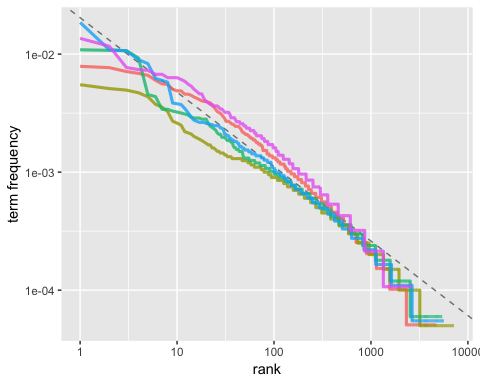
russia\_book\_words <- russia\_books %>%  
 unnest\_tokens(input=text,  
 output="word",  
 to\_lower=TRUE,  
 drop=TRUE) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 anti\_join(stop\_words, by=c("word"="word")) %>%  
 count(title, word, sort = TRUE) %>%  
 ungroup()  
  
russia\_total\_words <- russia\_book\_words %>%   
 group\_by(title) %>%   
 summarize(total = sum(n))  
  
russia\_book\_words <- left\_join(russia\_book\_words, russia\_total\_words)  
  
ggplot(russia\_book\_words, aes(n/total, fill = title)) +  
 geom\_histogram(show.legend = FALSE, bins = 25) +  
 xlim(NA, 0.0009) +  
 facet\_wrap(~title, ncol = 2, scales = "free\_y")



## Análise da comparação da frequência dos termos nos livros:

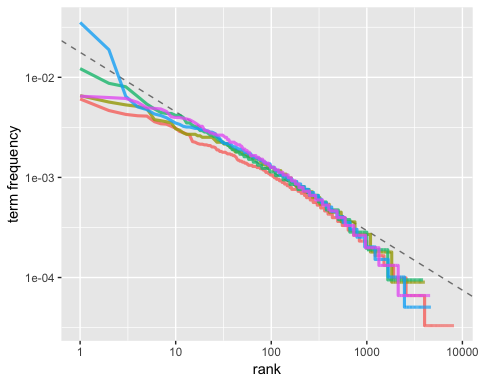
### França

france\_freq\_by\_rank <- france\_book\_words %>%   
 drop\_na() %>%  
 group\_by(title) %>%   
 mutate(rank = row\_number(),   
 `term frequency` = n/total)  
  
  
france\_rank\_subset <- france\_freq\_by\_rank %>%   
 filter(rank < 500,  
 rank > 10)  
  
france\_lm <- lm(log10(`term frequency`) ~ log10(rank), data = france\_rank\_subset)  
  
france\_freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = title)) +   
 geom\_abline(intercept = france\_lm$coefficients[1], slope = france\_lm$coefficients[2], color = "gray50", linetype = 2) +  
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



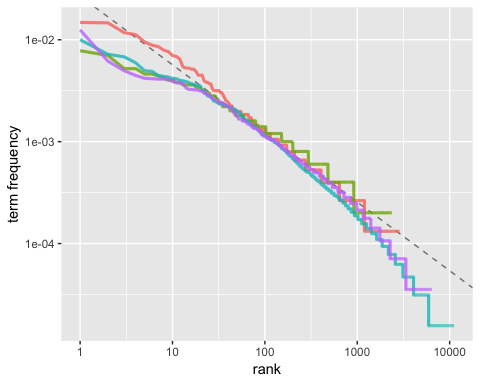
### Dardanelles:

dardanelles\_freq\_by\_rank <- dardanelles\_book\_words %>%   
 drop\_na() %>%  
 group\_by(title) %>%   
 mutate(rank = row\_number(),   
 `term frequency` = n/total)  
  
dardanelles\_rank\_subset <- dardanelles\_freq\_by\_rank %>%   
 filter(rank < 500,  
 rank > 10)  
  
dardanelles\_lm <- lm(log10(`term frequency`) ~ log10(rank), data = dardanelles\_rank\_subset)  
  
dardanelles\_freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = title)) +   
 geom\_abline(intercept = dardanelles\_lm$coefficients[1], slope = dardanelles\_lm$coefficients[2], color = "gray50", linetype = 2) +  
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



### Rússia:

russia\_freq\_by\_rank <- russia\_book\_words %>%   
 drop\_na() %>%  
 group\_by(title) %>%   
 mutate(rank = row\_number(),   
 `term frequency` = n/total)  
  
russia\_rank\_subset <- russia\_freq\_by\_rank %>%   
 filter(rank < 500,  
 rank > 10)  
  
russia\_lm <- lm(log10(`term frequency`) ~ log10(rank), data = russia\_rank\_subset)  
  
russia\_freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = title)) +   
 geom\_abline(intercept = russia\_lm$coefficients[1], slope = russia\_lm$coefficients[2], color = "gray50", linetype = 2) +  
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



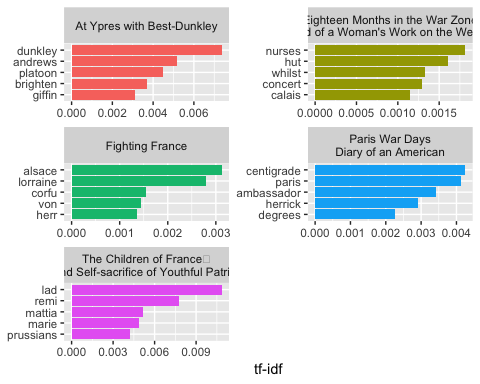
## TF-IDF (Term Frequency - Inverse Document Frequency)

### França

france\_book\_words <- france\_book\_words %>%  
 bind\_tf\_idf(word, title, n)  
  
france\_book\_words %>%  
 select(-total) %>%  
 arrange(desc(tf\_idf))

## # A tibble: 25,599 x 6  
## title word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 "The Children of France\r\nA Book … lad 63 0.00674 1.61 0.0108   
## 2 "The Children of France\r\nA Book … remi 45 0.00481 1.61 0.00775  
## 3 At Ypres with Best-Dunkley dunkley 90 0.00458 1.61 0.00736  
## 4 "The Children of France\r\nA Book … mattia 30 0.00321 1.61 0.00517  
## 5 At Ypres with Best-Dunkley andrews 63 0.00320 1.61 0.00515  
## 6 "The Children of France\r\nA Book … marie 50 0.00535 0.916 0.00490  
## 7 At Ypres with Best-Dunkley platoon 96 0.00488 0.916 0.00447  
## 8 "Paris War Days\nDiary of an Ameri… centig… 48 0.00264 1.61 0.00425  
## 9 "The Children of France\r\nA Book … prussi… 43 0.00460 0.916 0.00422  
## 10 "Paris War Days\nDiary of an Ameri… paris 337 0.0185 0.223 0.00413  
## # ... with 25,589 more rows

france\_book\_words %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(word, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()



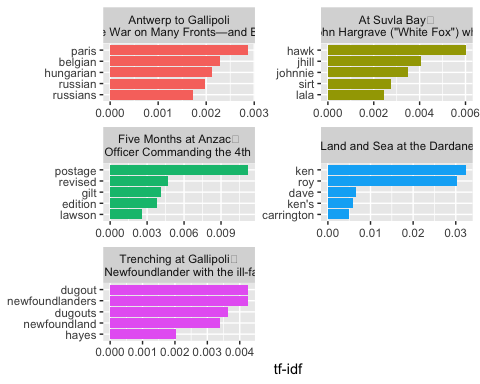
### Dardanelles

dardanelles\_book\_words <- dardanelles\_book\_words %>%  
 bind\_tf\_idf(word, title, n)  
  
dardanelles\_book\_words %>%  
 select(-total) %>%  
 arrange(desc(tf\_idf))

## # A tibble: 25,365 x 6  
## title word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 On Land and Sea at the Dardanell… ken 698 0.0353 0.916 0.0324   
## 2 On Land and Sea at the Dardanell… roy 373 0.0189 1.61 0.0304   
## 3 "Five Months at Anzac\r\nA Narra… postage 129 0.0122 0.916 0.0112   
## 4 On Land and Sea at the Dardanell… dave 82 0.00415 1.61 0.00668  
## 5 "At Suvla Bay\r\nBeing the notes… hawk 73 0.00656 0.916 0.00601  
## 6 On Land and Sea at the Dardanell… ken's 73 0.00369 1.61 0.00594  
## 7 On Land and Sea at the Dardanell… carringt… 60 0.00304 1.61 0.00489  
## 8 "Five Months at Anzac\r\nA Narra… revised 31 0.00294 1.61 0.00473  
## 9 "Trenching at Gallipoli\r\nThe p… dugout 40 0.00264 1.61 0.00426  
## 10 "Trenching at Gallipoli\r\nThe p… newfound… 40 0.00264 1.61 0.00426  
## # ... with 25,355 more rows

dardanelles\_book\_words %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(word, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()

## Selecting by tf\_idf

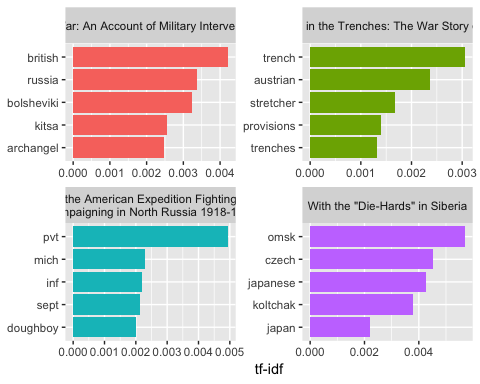


### Rússia:

russia\_book\_words <- russia\_book\_words %>%  
 bind\_tf\_idf(word, title, n)  
  
russia\_book\_words %>%  
 select(-total) %>%  
 arrange(desc(tf\_idf))

## # A tibble: 22,784 x 6  
## title word n tf idf tf\_idf  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 "With the \"Die-Hards\" in Siberia" omsk 116 0.00411 1.39 0.00570  
## 2 "The History of the American Exped… pvt 228 0.00356 1.39 0.00494  
## 3 "With the \"Die-Hards\" in Siberia" czech 92 0.00326 1.39 0.00452  
## 4 "With the \"Die-Hards\" in Siberia" japane… 174 0.00617 0.693 0.00427  
## 5 Fighting Without a War: An Account… british 111 0.0147 0.288 0.00421  
## 6 "With the \"Die-Hards\" in Siberia" koltch… 77 0.00273 1.39 0.00378  
## 7 Fighting Without a War: An Account… russia 89 0.0117 0.288 0.00338  
## 8 Fighting Without a War: An Account… bolshe… 85 0.0112 0.288 0.00323  
## 9 Four Weeks in the Trenches: The Wa… trench 22 0.00441 0.693 0.00305  
## 10 Fighting Without a War: An Account… kitsa 28 0.00370 0.693 0.00256  
## # ... with 22,774 more rows

russia\_book\_words %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(word, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()



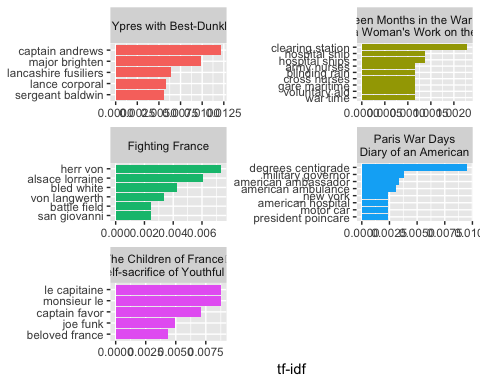
## Análise de Bigramas

### França:

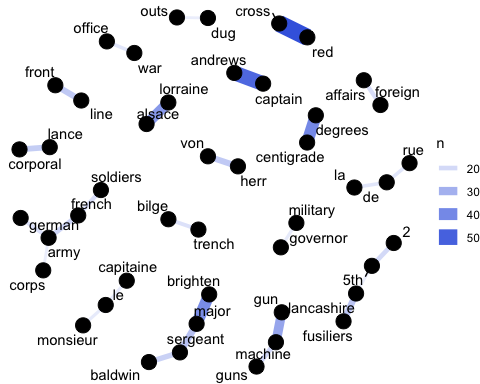
france\_bigrams <- france\_books %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
france\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 97,692 x 2  
## bigram n  
## <chr> <int>  
## 1 <NA> 5344  
## 2 of the 2217  
## 3 in the 1385  
## 4 to the 907  
## 5 on the 610  
## 6 at the 464  
## 7 and the 422  
## 8 for the 404  
## 9 to be 403  
## 10 it was 381  
## # ... with 97,682 more rows

france\_bigrams\_separated <- france\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
france\_bigrams\_filtered <- france\_bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
france\_bigram\_counts <- france\_bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
  
france\_bigrams\_united <- france\_bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
  
france\_bigram\_tf\_idf <- france\_bigrams\_united %>%  
 count(title, bigram) %>%  
 bind\_tf\_idf(bigram, title, n) %>%  
 arrange(desc(tf\_idf))  
  
france\_bigram\_tf\_idf %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(bigram, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()



france\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n >= 15) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n, edge\_width = n), edge\_colour = "royalblue") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,  
 point.padding = unit(0.2, "lines")) +  
 theme\_void()

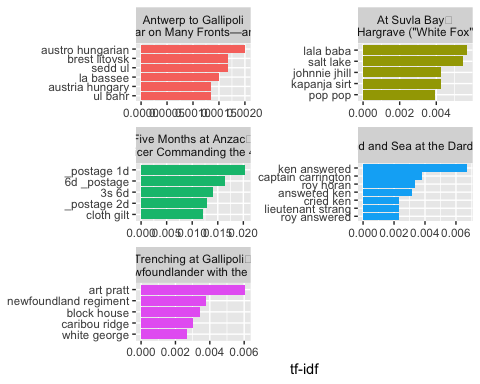


### Dardanelles:

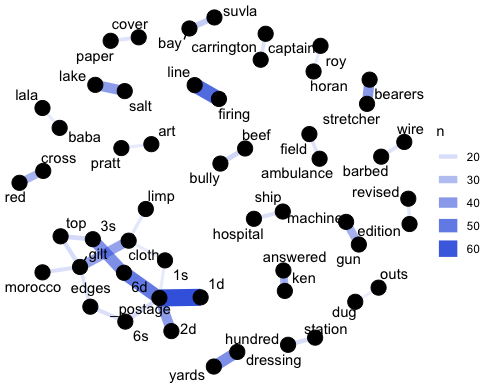
dardanelles\_bigrams <- dardanelles\_books %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
dardanelles\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 102,363 x 2  
## bigram n  
## <chr> <int>  
## 1 <NA> 6449  
## 2 of the 1783  
## 3 in the 1143  
## 4 to the 775  
## 5 and the 671  
## 6 on the 659  
## 7 it was 627  
## 8 in a 404  
## 9 from the 401  
## 10 at the 390  
## # ... with 102,353 more rows

dardanelles\_bigrams\_separated <- dardanelles\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
dardanelles\_bigrams\_filtered <- dardanelles\_bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
dardanelles\_bigram\_counts <- dardanelles\_bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
  
dardanelles\_bigrams\_united <- dardanelles\_bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
  
dardanelles\_bigram\_tf\_idf <- dardanelles\_bigrams\_united %>%  
 count(title, bigram) %>%  
 bind\_tf\_idf(bigram, title, n) %>%  
 arrange(desc(tf\_idf))  
  
dardanelles\_bigram\_tf\_idf %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(bigram, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()



dardanelles\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n >= 15) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n, edge\_width = n), edge\_colour = "royalblue") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,  
 point.padding = unit(0.2, "lines")) +  
 theme\_void()



### Rússia

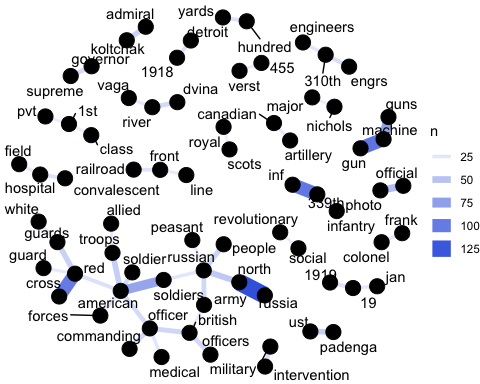
russia\_bigrams <- russia\_books %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
russia\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 109,418 x 2  
## bigram n  
## <chr> <int>  
## 1 <NA> 4220  
## 2 of the 2776  
## 3 in the 1371  
## 4 to the 1081  
## 5 on the 878  
## 6 and the 691  
## 7 it was 505  
## 8 for the 497  
## 9 by the 493  
## 10 had been 444  
## # ... with 109,408 more rows

russia\_bigrams\_separated <- russia\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
russia\_bigrams\_filtered <- russia\_bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
russia\_bigram\_counts <- russia\_bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
  
russia\_bigrams\_united <- russia\_bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
  
russia\_bigram\_tf\_idf <- russia\_bigrams\_united %>%  
 count(title, bigram) %>%  
 bind\_tf\_idf(bigram, title, n) %>%  
 arrange(desc(tf\_idf))  
  
russia\_bigram\_tf\_idf %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%   
 group\_by(title) %>%   
 top\_n(5) %>%   
 ungroup %>%  
 ggplot(aes(bigram, tf\_idf, fill = title)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~title, ncol = 2, scales = "free") +  
 coord\_flip()



russia\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n >= 20) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n, edge\_width = n), edge\_colour = "royalblue") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,  
 point.padding = unit(0.2, "lines")) +  
 theme\_void()



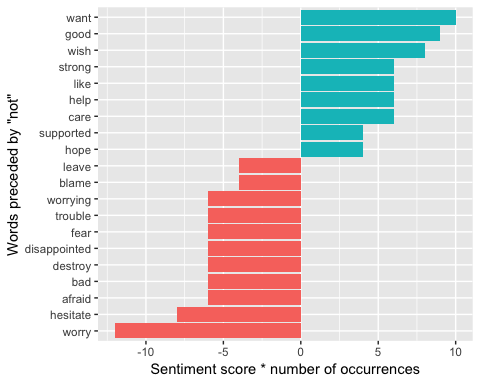
## Análise de Sentimentos dos Bigramas

### França

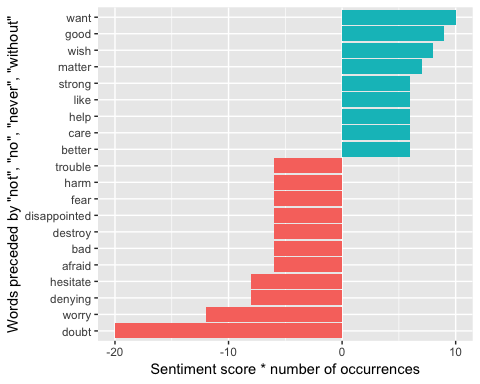
france\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 430 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not only 52  
## 2 not a 50  
## 3 not be 46  
## 4 not yet 32  
## 5 not been 30  
## 6 not know 29  
## 7 not to 27  
## 8 not the 25  
## 9 not get 17  
## 10 not have 15  
## # ... with 420 more rows

france\_not\_words <- france\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word2, score, sort = TRUE) %>%  
 ungroup()  
  
france\_not\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()



negation\_words <- c("not", "no", "never", "without")  
  
france\_negated\_words <- france\_bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word1, word2, score, sort = TRUE) %>%  
 ungroup()  
  
france\_negated\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\", \"no\", \"never\", \"without\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()

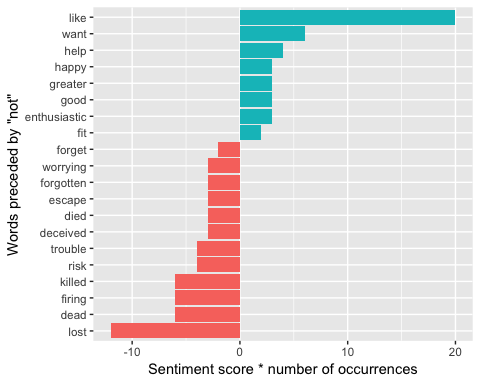


### Dardanelles

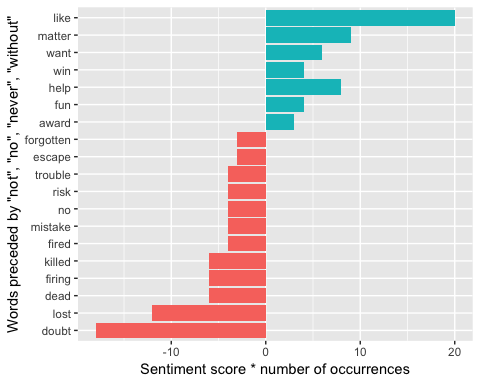
dardanelles\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 330 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not a 53  
## 2 not to 27  
## 3 not the 25  
## 4 not only 24  
## 5 not be 20  
## 6 not so 17  
## 7 not know 16  
## 8 not been 15  
## 9 not yet 15  
## 10 not get 14  
## # ... with 320 more rows

dardanelles\_not\_words <- dardanelles\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word2, score, sort = TRUE) %>%  
 ungroup()  
  
dardanelles\_not\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()



negation\_words <- c("not", "no", "never", "without")  
  
dardanelles\_negated\_words <- dardanelles\_bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word1, word2, score, sort = TRUE) %>%  
 ungroup()  
  
dardanelles\_negated\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\", \"no\", \"never\", \"without\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()

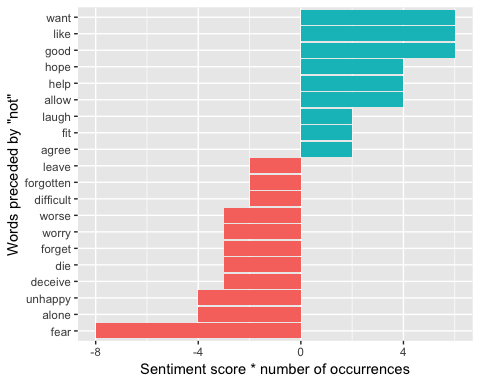


### Rússia:

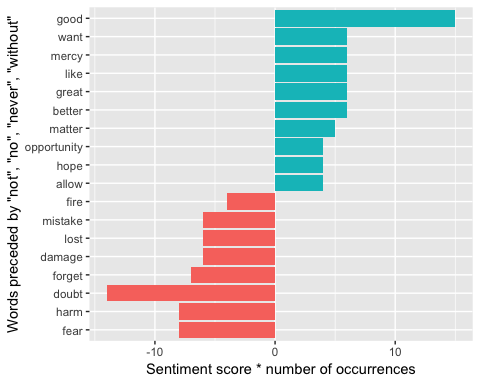
russia\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 411 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not be 40  
## 2 not a 38  
## 3 not to 35  
## 4 not the 32  
## 5 not know 24  
## 6 not only 23  
## 7 not been 22  
## 8 not have 19  
## 9 not one 18  
## 10 not see 16  
## # ... with 401 more rows

russia\_not\_words <- russia\_bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word2, score, sort = TRUE) %>%  
 ungroup()  
  
russia\_not\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()



negation\_words <- c("not", "no", "never", "without")  
  
russia\_negated\_words <- russia\_bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(get\_sentiments("afinn"), by = c(word2 = "word")) %>%  
 count(word1, word2, score, sort = TRUE) %>%  
 ungroup()  
  
russia\_negated\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\", \"no\", \"never\", \"without\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()



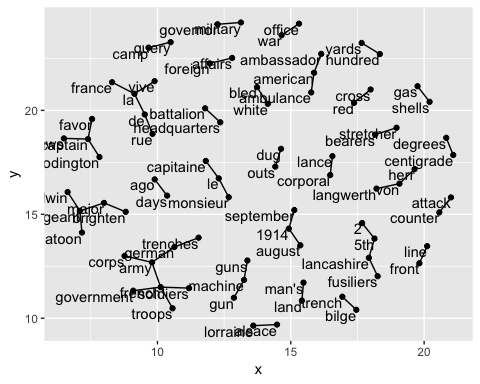
## Impressão de grafos dos bigramas:

### França

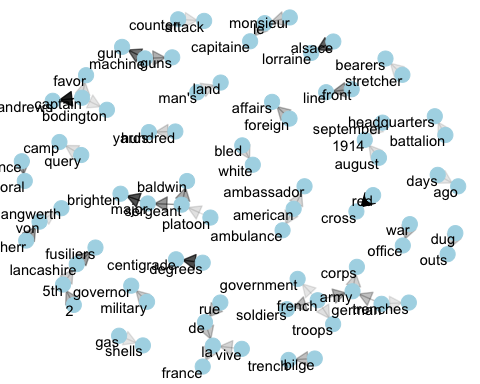
france\_bigram\_graph <- france\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n > 10) %>%  
 graph\_from\_data\_frame()  
  
france\_bigram\_graph

## IGRAPH 985ef5a DN-- 79 50 --   
## + attr: name (v/c), n (e/n)  
## + edges from 985ef5a (vertex names):  
## [1] red ->cross captain ->andrews degrees ->centigrade  
## [4] major ->brighten alsace ->lorraine machine ->gun   
## [7] sergeant ->major lancashire->fusiliers front ->line   
## [10] herr ->von lance ->corporal sergeant ->baldwin   
## [13] french ->army machine ->guns 2 ->5th   
## [16] foreign ->affairs french ->soldiers bilge ->trench   
## [19] german ->army de ->la rue ->de   
## [22] war ->office le ->capitaine military ->governor   
## + ... omitted several edges

set.seed(2018)  
  
ggraph(france\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)



a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(france\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()

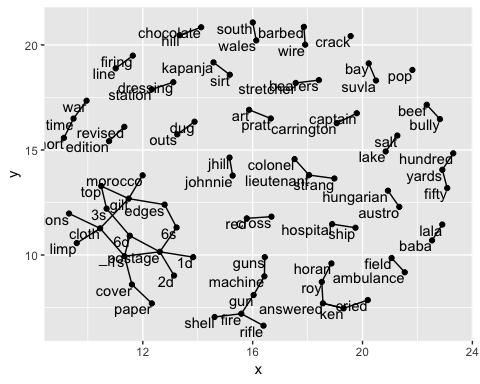


### Dardanelles:

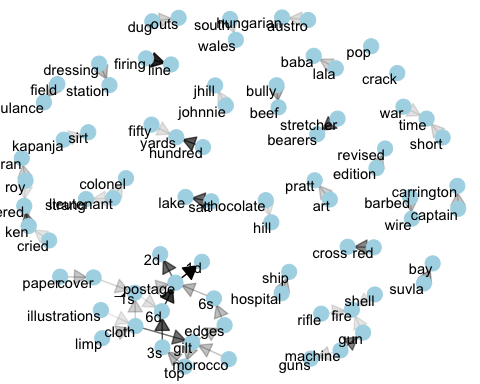
dardanelles\_bigram\_graph <- dardanelles\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n > 10) %>%  
 graph\_from\_data\_frame()  
  
dardanelles\_bigram\_graph

## IGRAPH 9c338de DN-- 78 57 --   
## + attr: name (v/c), n (e/n)  
## + edges from 9c338de (vertex names):  
## [1] \_postage ->1d firing ->line 6d ->\_postage   
## [4] 3s ->6d hundred ->yards stretcher->bearers   
## [7] \_postage ->2d salt ->lake cloth ->gilt   
## [10] machine ->gun ken ->answered red ->cross   
## [13] suvla ->bay bully ->beef gilt ->gilt   
## [16] top ->3s 6s ->\_postage dressing ->station   
## [19] gilt ->top edges ->6s hospital ->ship   
## [22] morocco ->gilt barbed ->wire captain ->carrington  
## + ... omitted several edges

set.seed(2018)  
  
ggraph(dardanelles\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)



a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(dardanelles\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()

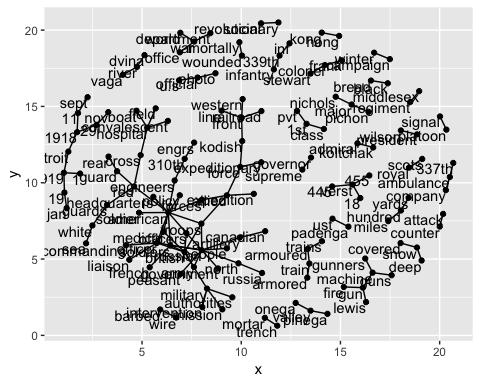


### Rússia:

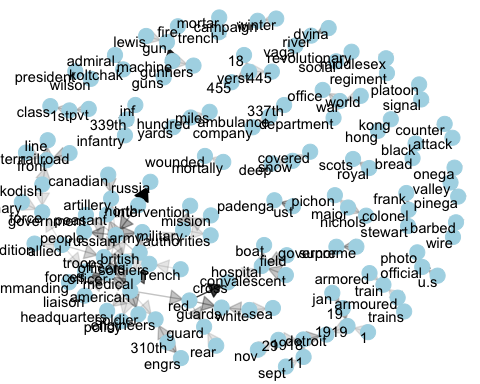
russia\_bigram\_graph <- russia\_bigram\_counts %>%  
 drop\_na() %>%  
 filter(n > 10) %>%  
 graph\_from\_data\_frame()  
  
russia\_bigram\_graph

## IGRAPH 09419cd DN-- 138 117 --   
## + attr: name (v/c), n (e/n)  
## + edges from 09419cd (vertex names):  
## [1] north ->russia machine ->gun 339th ->inf   
## [4] red ->cross machine ->guns american ->soldiers  
## [7] official ->photo north ->russian american ->troops   
## [10] ust ->padenga red ->guards admiral ->koltchak  
## [13] russian ->people british ->officer russian ->army   
## [16] british ->officers commanding->officer dvina ->river   
## [19] american ->officer american ->soldier supreme ->governor  
## [22] 310th ->engrs   
## + ... omitted several edges

set.seed(2018)  
  
ggraph(russia\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)



a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(russia\_bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()



## Correlação dos Bigramas

### França:

france\_section\_words <- france\_books %>%  
 mutate(section = row\_number() %/% 10) %>%  
 filter(section > 0) %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 filter(!word %in% stop\_words$word)  
  
france\_section\_words

## # A tibble: 83,918 x 5  
## gutenberg\_id title linenumber section word   
## <int> <chr> <int> <dbl> <chr>   
## 1 9975 "Paris War Days\nDiary of an A… 11 1 diary   
## 2 9975 "Paris War Days\nDiary of an A… 11 1 americ…  
## 3 9975 "Paris War Days\nDiary of an A… 19 1 charles  
## 4 9975 "Paris War Days\nDiary of an A… 19 1 inman   
## 5 9975 "Paris War Days\nDiary of an A… 19 1 barnard  
## 6 9975 "Paris War Days\nDiary of an A… 19 1 ll   
## 7 9975 "Paris War Days\nDiary of an A… 19 1 harvard  
## 8 9975 "Paris War Days\nDiary of an A… 21 2 knight   
## 9 9975 "Paris War Days\nDiary of an A… 21 2 legion   
## 10 9975 "Paris War Days\nDiary of an A… 21 2 honor   
## # ... with 83,908 more rows

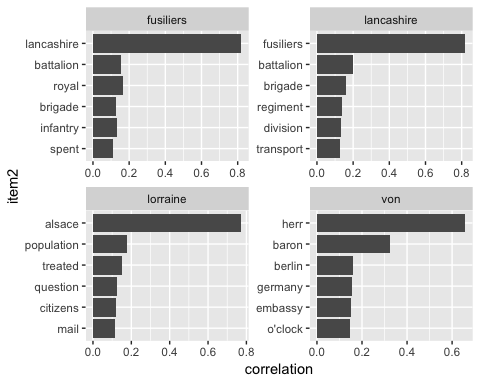
france\_word\_pairs <- france\_section\_words %>%  
 pairwise\_count(word, section, sort = TRUE) %>%  
 drop\_na()  
  
france\_word\_pairs

## # A tibble: 2,042,368 x 3  
## item1 item2 n  
## <chr> <chr> <dbl>  
## 1 french german 119  
## 2 german french 119  
## 3 french war 105  
## 4 war french 105  
## 5 day war 104  
## 6 war day 104  
## 7 france war 92  
## 8 war france 92  
## 9 german war 89  
## 10 war german 89  
## # ... with 2,042,358 more rows

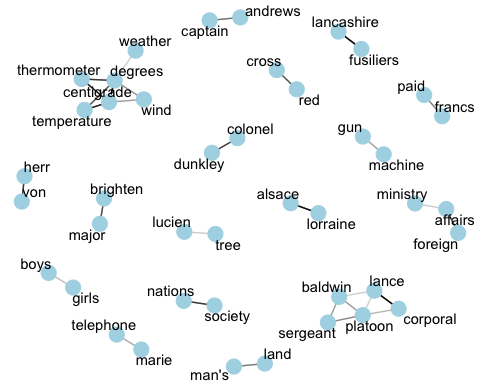
france\_word\_cors <- france\_section\_words %>%  
 na.omit() %>%  
 group\_by(word) %>%  
 filter(n() >= 20) %>%  
 pairwise\_cor(word, section, sort = TRUE)  
  
france\_word\_cors

## # A tibble: 652,056 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 centigrade degrees 0.906  
## 2 degrees centigrade 0.906  
## 3 lancashire fusiliers 0.820  
## 4 fusiliers lancashire 0.820  
## 5 corporal lance 0.806  
## 6 lance corporal 0.806  
## 7 lorraine alsace 0.773  
## 8 alsace lorraine 0.773  
## 9 centigrade thermometer 0.704  
## 10 thermometer centigrade 0.704  
## # ... with 652,046 more rows

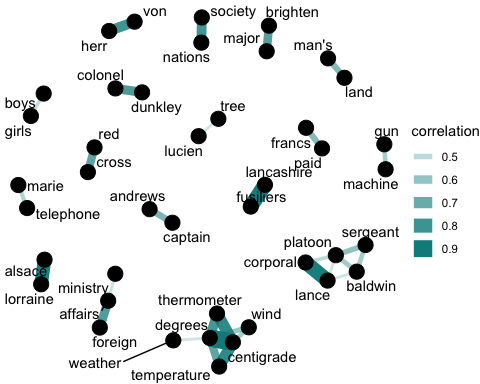
france\_word\_cors %>%  
 filter(item1 %in% c("lorraine", "lancashire", "fusiliers", "von")) %>%  
 group\_by(item1) %>%  
 top\_n(6) %>%  
 ungroup() %>%  
 mutate(item2 = reorder(item2, correlation)) %>%  
 ggplot(aes(item2, correlation)) +  
 geom\_bar(stat = "identity") +  
 facet\_wrap(~ item1, scales = "free") +  
 coord\_flip()



set.seed(2018)  
  
france\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation), show.legend = FALSE) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_void()



france\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation, edge\_width = correlation), edge\_colour = "cyan4") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,   
 point.padding = unit(0.2, "lines")) +  
 theme\_void()



### Dardanelles:

dardanelles\_section\_words <- dardanelles\_books %>%  
 mutate(section = row\_number() %/% 10) %>%  
 filter(section > 0) %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 filter(!word %in% stop\_words$word)  
  
dardanelles\_section\_words

## # A tibble: 86,828 x 5  
## gutenberg\_id title linenumber section word   
## <int> <chr> <int> <dbl> <chr>   
## 1 3306 "At Suvla Bay\r\nBeing the not… 11 1 serving  
## 2 3306 "At Suvla Bay\r\nBeing the not… 11 1 field   
## 3 3306 "At Suvla Bay\r\nBeing the not… 11 1 ambula…  
## 4 3306 "At Suvla Bay\r\nBeing the not… 11 1 divisi…  
## 5 3306 "At Suvla Bay\r\nBeing the not… 11 1 medite…  
## 6 3306 "At Suvla Bay\r\nBeing the not… 12 1 expedi…  
## 7 3306 "At Suvla Bay\r\nBeing the not… 12 1 force   
## 8 3306 "At Suvla Bay\r\nBeing the not… 12 1 war   
## 9 3306 "At Suvla Bay\r\nBeing the not… 18 1 minobi   
## 10 3306 "At Suvla Bay\r\nBeing the not… 20 2 played   
## # ... with 86,818 more rows

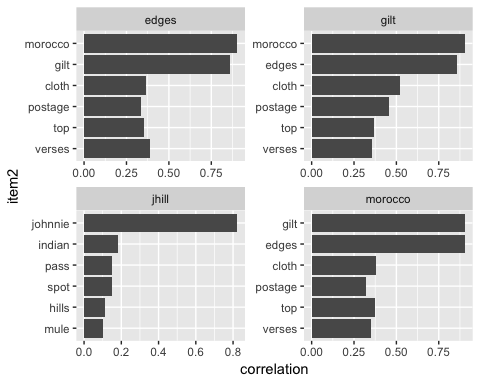
dardanelles\_word\_pairs <- dardanelles\_section\_words %>%  
 pairwise\_count(word, section, sort = TRUE) %>%  
 drop\_na()  
  
dardanelles\_word\_pairs

## # A tibble: 2,169,572 x 3  
## item1 item2 n  
## <chr> <chr> <dbl>  
## 1 roy ken 215  
## 2 ken roy 215  
## 3 ken time 83  
## 4 time ken 83  
## 5 ken answered 80  
## 6 answered ken 80  
## 7 postage cloth 76  
## 8 cloth postage 76  
## 9 ken moment 67  
## 10 moment ken 67  
## # ... with 2,169,562 more rows

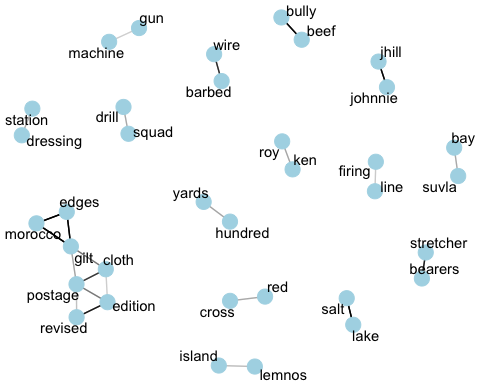
dardanelles\_word\_cors <- dardanelles\_section\_words %>%  
 na.omit() %>%  
 group\_by(word) %>%  
 filter(n() >= 20) %>%  
 pairwise\_cor(word, section, sort = TRUE)  
  
dardanelles\_word\_cors

## # A tibble: 818,120 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 morocco gilt 0.905  
## 2 gilt morocco 0.905  
## 3 edges morocco 0.904  
## 4 morocco edges 0.904  
## 5 lake salt 0.876  
## 6 salt lake 0.876  
## 7 edges gilt 0.859  
## 8 gilt edges 0.859  
## 9 beef bully 0.836  
## 10 bully beef 0.836  
## # ... with 818,110 more rows

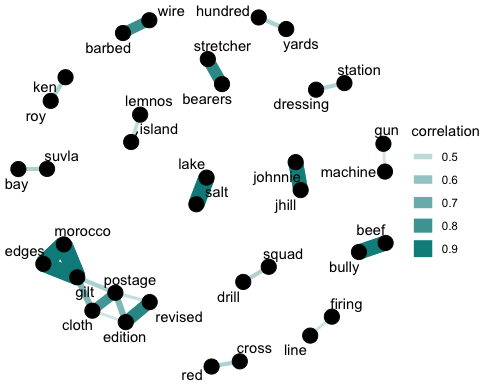
dardanelles\_word\_cors %>%  
 filter(item1 %in% c("edges", "morocco", "jhill", "gilt")) %>%  
 group\_by(item1) %>%  
 top\_n(6) %>%  
 ungroup() %>%  
 mutate(item2 = reorder(item2, correlation)) %>%  
 ggplot(aes(item2, correlation)) +  
 geom\_bar(stat = "identity") +  
 facet\_wrap(~ item1, scales = "free") +  
 coord\_flip()



set.seed(2018)  
  
dardanelles\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation), show.legend = FALSE) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_void()



dardanelles\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation, edge\_width = correlation), edge\_colour = "cyan4") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,   
 point.padding = unit(0.2, "lines")) +  
 theme\_void()



### Rússia:

russia\_section\_words <- russia\_books %>%  
 mutate(section = row\_number() %/% 10) %>%  
 filter(section > 0) %>%  
 unnest\_tokens(word, text) %>%  
 mutate(word=str\_extract(word,"[a-z']+")) %>%  
 filter(!word %in% stop\_words$word)  
  
russia\_section\_words

## # A tibble: 104,754 x 5  
## gutenberg\_id title linenumber section word   
## <int> <chr> <int> <dbl> <chr>   
## 1 10967 Four Weeks in the Trenches: Th… 10 1 life   
## 2 10967 Four Weeks in the Trenches: Th… 10 1 dedica…  
## 3 10967 Four Weeks in the Trenches: Th… 10 1 book   
## 4 10967 Four Weeks in the Trenches: Th… 12 1 humble   
## 5 10967 Four Weeks in the Trenches: Th… 12 1 token   
## 6 10967 Four Weeks in the Trenches: Th… 12 1 everla…  
## 7 10967 Four Weeks in the Trenches: Th… 12 1 gratit…  
## 8 10967 Four Weeks in the Trenches: Th… 12 1 devoti…  
## 9 10967 Four Weeks in the Trenches: Th… 17 1 preface  
## 10 10967 Four Weeks in the Trenches: Th… 19 1 record   
## # ... with 104,744 more rows

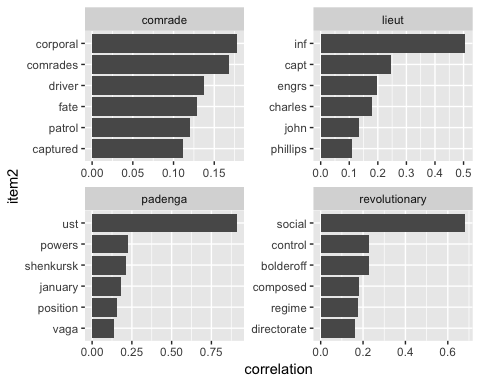
russia\_word\_pairs <- russia\_section\_words %>%  
 pairwise\_count(word, section, sort = TRUE) %>%  
 drop\_na()  
  
russia\_word\_pairs

## # A tibble: 2,576,030 x 3  
## item1 item2 n  
## <chr> <chr> <dbl>  
## 1 russian american 167  
## 2 american russian 167  
## 3 british american 146  
## 4 american british 146  
## 5 british russian 141  
## 6 north russia 141  
## 7 russia north 141  
## 8 russian british 141  
## 9 archangel american 138  
## 10 american archangel 138  
## # ... with 2,576,020 more rows

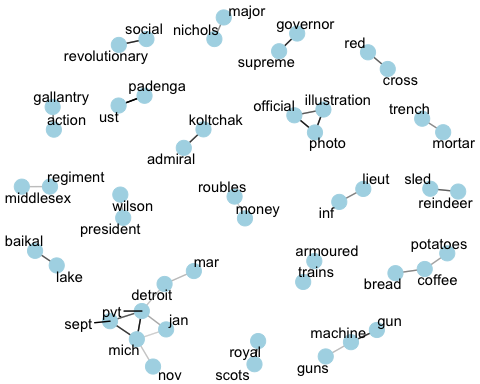
russia\_word\_cors <- russia\_section\_words %>%  
 na.omit() %>%  
 group\_by(word) %>%  
 filter(n() >= 20) %>%  
 pairwise\_cor(word, section, sort = TRUE)  
  
russia\_word\_cors

## # A tibble: 1,097,256 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 padenga ust 0.910  
## 2 ust padenga 0.910  
## 3 pvt mich 0.734  
## 4 mich pvt 0.734  
## 5 photo official 0.724  
## 6 official photo 0.724  
## 7 photo illustration 0.723  
## 8 illustration photo 0.723  
## 9 governor supreme 0.686  
## 10 supreme governor 0.686  
## # ... with 1,097,246 more rows

russia\_word\_cors %>%  
 filter(item1 %in% c("padenga", "revolutionary", "comrade", "lieut")) %>%  
 group\_by(item1) %>%  
 top\_n(6) %>%  
 ungroup() %>%  
 mutate(item2 = reorder(item2, correlation)) %>%  
 ggplot(aes(item2, correlation)) +  
 geom\_bar(stat = "identity") +  
 facet\_wrap(~ item1, scales = "free") +  
 coord\_flip()



set.seed(2018)  
  
russia\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation), show.legend = FALSE) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_void()



russia\_word\_cors %>%  
 filter(correlation > .4) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation, edge\_width = correlation), edge\_colour = "cyan4") +  
 geom\_node\_point(size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE,   
 point.padding = unit(0.2, "lines")) +  
 theme\_void()

