Text Mining

require(tidyverse)

library(foreign)

# Load Texts

|  |  |  |
| --- | --- | --- |
| dokument | category | text |
| … | … | … |

list files

files = list.files(recursive = TRUE, pattern = "\*.txt")

getDocument = function(x){

extract words

paste(unlist(scan(x, what="character",quiet = T)),collapse = " ")

}

text = map\_chr(files, getDocument)

create DF

textData = data.frame(document=1:2225, text=text)

files %>%

str\_remove\_all("\\d") %>%

add categories

str\_remove\_all('/.txt')-> categories

textData$category = categories

textData$text = as.character(textData$text)

# Term Frequencies

|  |  |  |
| --- | --- | --- |
| Category | Word | N ( = TF) |
| … | … | … |

textData %>%

group\_by(category) %>%

unnest\_tokens(output = word, input = text) %>%

anti\_join(stop\_words) %>%

count(word, sort = TRUE) -> termFrequencies

In wie vielen Dokumenten kommt Wort vor?

log(N/df)

TF \* IDF

Wie oft kommt Wort in einem Dokument vor?

# TF-IDF



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Category | Word | N | TF | IDF | TF-IDF |
| … | … | … | … | … | … |

textData %>%

unnest\_tokens(word, text) %>%

anti\_join(stop\_words) %>%

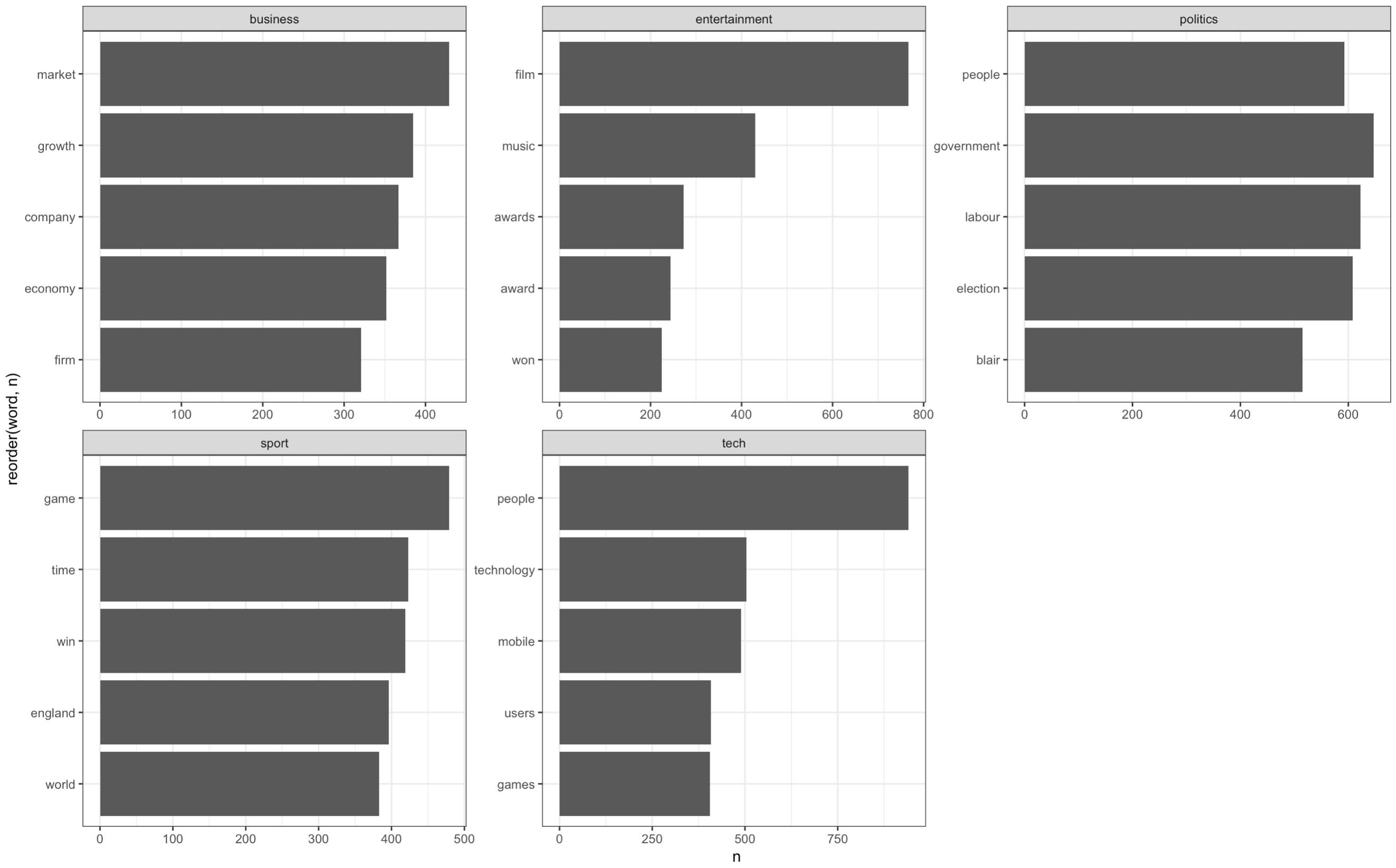
count(category, word, sort = TRUE) %>%

bind\_tf\_idf(word, category, n) %>%

group\_by(category) %>%

arrange(desc(tf\_idf)) -> inverseTermFrequencies

# Plot Top Terms



data %>%

top\_n(5) %>%

ggplot(aes(x=reorder(word, n), y=n)) +

geom\_col(position="dodge") +

facet\_wrap(.~category, scales = 'free') +

coord\_flip() +

theme\_bw()

# LDA Classification

library(topicmodels)

## Term-Document-Matrix

textData %>%

unnest\_tokens(**output =** word, **input =** text) %>%

anti\_join(stop\_words) %>%

count(document, word, sort = TRUE) %>%

cast\_dtm(document, **term =** word, **value =** n) -> tdm



## LDA Model

lda = LDA(tdm,

k=length(unique(categories)),

Create LDA Model

control = list(seed=1234))

ldaTD = tidy(lda)

ldaTD %>%

Topic durch LDA Modell erstellt

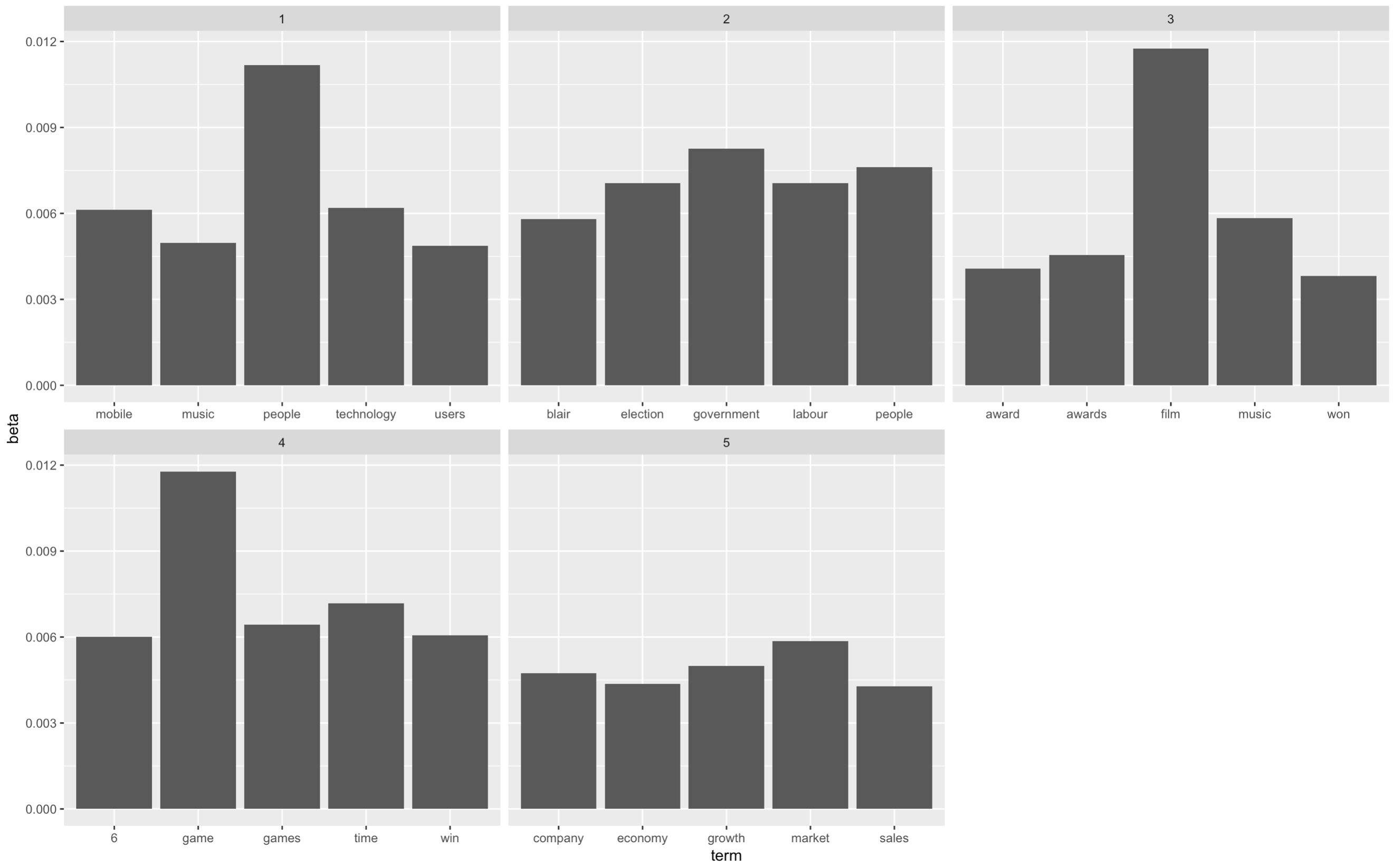
group\_by(topic) %>%

Finde die 5 besten Terme zur Klassifizierung

top\_n(5, beta) %>%

arrange(topic, desc(beta)) -> topTerms

## 5.3 Plot Top 5 Terms

topTerms %>%

mutate(term = reorder\_within(term, beta, topic)) %>%

ggplot(aes(term, beta)) +

geom\_bar(stat = "identity") +

scale\_x\_reordered() +

facet\_wrap(~ topic, scales = "free\_x")

## 5.4 Predict Classification based on Words

ldaGamma = tidy(lda, matrix = 'gamma')

ldaGamma %>%

arrange(document) %>%

group\_by(document) %>%

top\_n(1, gamma) %>%

ungroup() %>%

arrange(gamma) %>%

mutate(document = as.integer(document)) %>%

mutate(topic = if\_else(topic == 1, "tech" ,

if\_else(topic == 2, "politics",

if\_else(topic == 3, "entertainment",

if\_else(topic == 4, "sport", "business"))))) -> docClass

## 5.5 Evaluate Classification Results

textData %>%

Merge Text Data and

Classification Data

left\_join(documentClassifications, by = "document") %>%

mutate(correctLabel = category == topic) -> textDataClassified

Share of correct labled data

mean(textDataClassified$correctLabel)