## Пример классификации текстовых документов

### Классификация пресс-релизов британского правительства

Пресс-релизы британского правительства находятся по следующей ссылке:

[https://www.gov.uk/government/announcements?keywords=&announcement\_type\_option=press-releases&topics[]=all&departments[]=all&world\_locations[]=all&from\_date=&to\_date=01%2F07%2F201**9**](https://www.gov.uk/government/announcements?keywords=&announcement_type_option=press-releases&topics%5b%5d=all&departments%5b%5d=all&world_locations%5b%5d=all&from_date=&to_date=01%2F07%2F2019)

# 

#### Необходимые пакеты:

library("stringr")

library ("RCurl")

library ("XML")

Проанализируем HTML–код. Каждый пресс-релиз содержится в гиперссылке, которая содержится в теге **<li>** с атрибутом **id**.

<li id="news\_article\_560756" class="document-row">

<h3><a href="[/government/news/foreign-secretary-congratulates-sierra-leone-on-eliminating-ebola](view-source:https://www.gov.uk/government/news/foreign-secretary-congratulates-sierra-leone-on-eliminating-ebola)">Foreign Secretary congratulates Sierra Leone on eliminating Ebola</a></h3>

<ul class="attributes">

<li><time class="public\_timestamp" datetime="2015-11-07T10:47:47+00:00"> 7 November 2015</time></li>

<li class="organisations"><abbr title="Foreign &amp; Commonwealth Office">FCO</abbr></li>

<li class="display-type">Press release</li>

</ul>

</li>

Для перехода на следующую страницу нужно извлечь гиперссылку из тега

<li class="next">, который находится в теге <nav id="show-more-documents">

<nav id="show-more-documents" role="navigation">

<ul class="previous-next-navigation">

<li class="next">

<a href="[/government/announcements?announcement\_filter\_option=press-releases&amp;departments%5B%5D=all&amp;from\_date=01%2F01%2F2015&amp;keywords=&amp;page=2&amp;to\_date=&amp;topics%5B%5D=all](view-source:https://www.gov.uk/government/announcements?announcement_filter_option=press-releases&departments%5B%5D=all&from_date=01%2F01%2F2015&keywords=&page=2&to_date=&topics%5B%5D=all)">Next <span class="visuallyhidden">page</span> <span class="page-numbers">2 of 93</span></a>

</li>

</ul>

</nav>

# # Зададим временной интервал до июля 2018 г.

new\_results <- '/government/announcements?keywords=&announcement\_filter\_option=press-releases&topics[]=all&departments[]=all&world\_locations[]=all& from\_date=&to\_date=01%2F07%2F2018'

# # создаем объект для работы по протоколу SSL

signatures = system.file("CurlSSL", cainfo = "cacert.pem",

package = "RCurl")

all\_links <- character()

# # Перебираем страницы

while(length(new\_results) > 0)

{

new\_results <- str\_c("https://www.gov.uk/", new\_results)

results <- getURL(new\_results, cainfo = signatures)

results\_tree <- htmlParse(results)

# # берем со страницы все гиперссылки на пресс-релизы

all\_links <- c(all\_links, xpathSApply(results\_tree,

"//li[@id]//a", xmlGetAttr, "href"))

# # новая страница для загрузки

new\_results <- xpathSApply(results\_tree,

"//nav[@id='show-more-documents']//li[@class='next']//a",

xmlGetAttr, "href")

}

# # Теперь загружаем все страницы на диск

for(i in 1:length(all\_links)){

url <- str\_c("https://www.gov.uk", all\_links[i])

tmp <- getURL(url, cainfo = signatures)

write(tmp, str\_c("Press\_Releases/", i, ".html"))

}

# # Читаем первый пресс-релиз

tmp <- readLines("Press\_Releases/1.html")

# # Убираем переносы строк

tmp <- str\_c(tmp, collapse = "")

# # Строим HTML-объект

tmp <- htmlParse(tmp)

#### Текст пресс-релиза:

<div class="block-4">

<div class="inner-block">

<div class="document body">

<div class="govspeak">

<p>A government spokesperson said:</p>

<blockquote>

<p>Following the earlier suspension of flights from Sharm el-Sheikh airport, we continue intensive work to allow UK nationals to return to the UK in the coming days. Around 1,500 people returned to the UK yesterday and we expect similar numbers to return today.</p>

<p>We are working closely with the Egyptian authorities, UK airlines and tour operators to have passengers on flights and returned to the UK as soon as possible. We are grateful to all parties for the work they are doing in this difficult situation, in particular the airlines and operators for their efforts on behalf of their customers.</p>

<p>With a limited number of flights able to leave Sharm el-Sheikh each day for the UK, it is likely that tour operators or airlines will advise some people to extend their stay at their resort.</p>

<p>We understand that tour operators and airlines are working to ensure that where people need to extend their stay at their resort necessary costs will be covered.</p>

<p>It is important that people stay at their resort until they have confirmation from their airline or tour operator that they are on a flight back to the UK and that they follow their airline’s advice on the appropriate arrival time at the airport.</p>

<p>The additional security measures we have put in place have led to delays to flights, and we understand the frustration that has caused, but our top priority will continue to be the safety of British nationals returning to the UK.</p>

<p>Please continue to liaise with your airline and travel operator over the coming days. The situation remains fluid.</p>

<p class="last-child">Any British nationals requiring assistance, over and above that being provided by their airline and tour operators, can call the Foreign and Commonwealth Office on 020 7008 0000.</p>

</blockquote>

#### Информация об организации:

<div class="inner-heading">

<dl>

<dt>From:</dt>

<dd class="js-hide-extra-metadata">

<a class="organisation-link" href="[/government/organisations/prime-ministers-office-10-downing-street](view-source:https://www.gov.uk/government/organisations/prime-ministers-office-10-downing-street)">Prime Minister's Office, 10 Downing Street</a>, <a class="organisation-link" href="[/government/organisations/foreign-commonwealth-office](view-source:https://www.gov.uk/government/organisations/foreign-commonwealth-office)">Foreign &amp; Commonwealth Office</a> and <a class="organisation-link" href="[/government/organisations/department-for-transport](view-source:https://www.gov.uk/government/organisations/department-for-transport)">Department for Transport</a>

</dd>

#### Дата публикации:

<div class="block-5">

<div class="inner-block">

<div class="document-footer-meta js-footer">

<div id="history" class="history">

<dl class="primary-metadata">

<dt>Published:</dt>

<dd><time class="date" datetime="2015-11-07T17:21:42+00:00"> 7 November 2015</time></dd>

</dl>

</div>

release <- xpathSApply(tmp, "//div[@class='block-4']", xmlValue)

organisation <- xpathSApply(tmp, "//a[@class='organisation-link']", xmlValue)

publication <- xpathSApply(tmp, "//div[@class='block-5']//time[@class='date']", xmlValue)

# # пакет Text mining

install.packages("tm")

library(tm)

# # Создаем объект корпус

release\_corpus <- Corpus(VectorSource(release))

# # Добавляем метаинформацию

meta(release\_corpus[[1]], "organisation") <- organisation[1]

meta(release\_corpus[[1]], "publication") <- publication

meta(release\_corpus[[1]])

author : character(0)

datetimestamp: 2015-11-08 16:53:15

description : character(0)

heading : character(0)

id : 1

language : en

origin : character(0)

organisation : Prime Minister's Office, 10 Downing Street

publication : 24 August 2007

# # Добавляем в корпус все остальные документы (выбрали 3 министерства)

n <- 1

for(i in 2:length(list.files("Press\_Releases/"))){

tmp <- readLines(str\_c("Press\_Releases/", i, ".html"))

tmp <- str\_c(tmp, collapse = "")

tmp <- htmlParse(tmp)

release <- xpathSApply(tmp,"//div[@class='block-4']", xmlValue)

organisation <- xpathSApply(tmp, "//a[@class='organisation-link']", xmlValue)

publication <- xpathSApply(tmp, "//div[@class='block-5']//time[@class='date']", xmlValue)

if (length(release)!=0 &

(organisation == 'Department for Business, Innovation & Skills' |

organisation == 'Ministry of Defence' |

organisation == 'Foreign & Commonwealth Office')) {

n <- n + 1

tmp\_corpus <- Corpus(VectorSource(release))

release\_corpus <- c(release\_corpus, tmp\_corpus)

meta(release\_corpus[[n]], "organisation") <- organisation[1]

cat("n=",n)

}

}

release\_corpus

<<VCorpus>>

Metadata: corpus specific: 0, document level (indexed): 0

Content: documents: 446

# Список министерств, выпустивших пресс-релизы

meta\_data<- data.frame()

for (i in 1:NROW(release\_corpus))

{

meta\_data [i, "organisation"] <- meta(release\_corpus[[i]], "organisation")

meta\_data [i, "num"] <- i

}

table(as.character(meta\_data[, "organisation"])

Department for Business, Innovation & Skills

65

Foreign & Commonwealth Office

204

Ministry of Defence

177

# Теперь выполняем преобразования данных (функция content\_transformer применяется для того,чтобы не менялся тип данных)

# Удаляем числа

release\_corpus <- tm\_map(release\_corpus, content\_transformer(removeNumbers))

# Знаки пунктуации заменяем на пробелы

release\_corpus <- tm\_map(release\_corpus,

content\_transformer(str\_replace\_all),

pattern = "[[:punct:]]", replacement = " ")

release\_corpus[[1]]$content

# Удаляем стоп-слова

release\_corpus <- tm\_map(release\_corpus, content\_transformer(removeWords), words = stopwords("en"))

# Преобразуем к нижнему регистру

release\_corpus <- tm\_map(release\_corpus, content\_transformer(tolower))

# Проводим стемминг

release\_corpus <- tm\_map(release\_corpus, stemDocument, language = "english")

# Создаем матрицу терминов-документов

tdm <- TermDocumentMatrix(release\_corpus)

tdm

<<TermDocumentMatrix (terms: 7758, documents: 446)>>

Non-/sparse entries: 60624/3399444

Sparsity : 98%

Maximal term length: 26

Weighting : term frequency (tf)

# Можно фильтровать документы по ключевым словам

(nucl <- tm\_filter(release\_corpus,

FUN = function(x) any(str\_detect(x, "nuclear"))))

<<VCorpus>>

Metadata: corpus specific: 0, document level (indexed): 0

Content: documents: 27

# Можно отыскивать взаимосвязанные термины

findAssocs(tdm, "nuclear", .9)

$nuclear

weapon disarma treati

0.94 0.91 0.91

# Создаем матрицу документов-терминов

dtm <- DocumentTermMatrix(release\_corpus)

dtm

<<DocumentTermMatrix (documents: 446, terms: 7758)>>

Non-/sparse entries: 60624/3399444

Sparsity : 98%

Maximal term length: 26

Weighting : term frequency (tf)

dtm <- removeSparseTerms(dtm, 1-(10/length(release\_corpus)))

dtm

<<DocumentTermMatrix (documents: 446, terms: 1374)>>

Non-/sparse entries: 46354/566450

Sparsity : 92%

Maximal term length: 22

Weighting : term frequency (tf)

# Дополнительный пакет для работы с текстами

install.packages("RTextTools")

library(RTextTools)

# Массив меток – названий министерств

org\_labels<-meta\_data[, "organisation"]

N <- length(org\_labels)

# Для применения методов классификации создаем контейнер

container <- create\_container(

dtm,

labels = org\_labels,

trainSize = 1:350,

testSize = 351:N,

virgin = FALSE

)

# Запускаем обучение моделей

# Support vector machine

svm\_model <- train\_model(container, "SVM")

# Random forest

tree\_model <- train\_model(container, "TREE")

# Maximal entropy

maxent\_model <- train\_model(container, "MAXENT")

# Запускаем классификацию

svm\_out <- classify\_model(container, svm\_model)

tree\_out <- classify\_model(container, tree\_model)

maxent\_out <- classify\_model(container, maxent\_model)

# Создаем общую матрицу меток

labels\_out <- data.frame(

correct\_label = org\_labels[351:N],

svm = as.character(svm\_out[,1]),

tree = as.character(tree\_out[,1]),

maxent = as.character(maxent\_out[,1]),

stringsAsFactors = F)

## SVM performance

table(labels\_out[,1] == labels\_out[,2])

FALSE TRUE

9 87

## Random forest performance

table(labels\_out[,1] == labels\_out[,3])

FALSE TRUE

19 77

## Maximum entropy performance

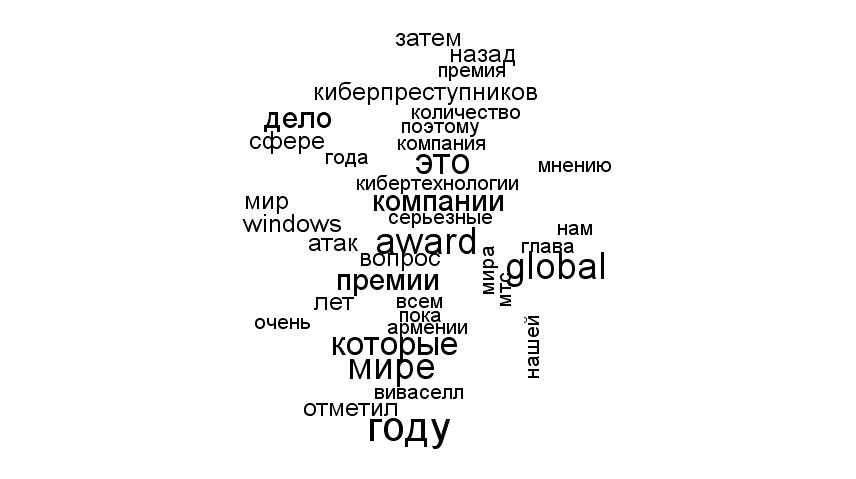
table(labels\_out[,1] == labels\_out[,4])

FALSE TRUE

9 87

## Стемминг

#### Создание облака слов без стемминга



#### Создание облака слов с обычным стеммингом из пакета tm (слова сокращаются)



#### Создание облака слов со стеммингом от Яндекс

<https://tech.yandex.ru/mystem/doc/index-docpage/>

<http://matrunich.com/blog/2014/05/12/wordcloud/>



## Анализ тональности (sentiment analysis). С чего начать?

При использовании подхода, основанного на словарях, загружаем словари, например:

<http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>

pos <- readLines("positive-words.txt")

# удаляем комментарии

pos <- pos[!str\_detect(pos, ";")]

pos <- pos[2:length(pos)]

neg <- readLines("negative-words.txt")

# удаляем комментарии

neg <- neg[!str\_detect(neg, ";")]

neg <- neg[2:length(neg)]

pos[1:100]

[1] "a+" "abound" "abounds" "abundance" "abundant"

[6] "accessable" "accessible" "acclaim" "acclaimed" "acclamation"

[11] "accolade" "accolades" "accommodative" "accomodative" "accomplish"

[16] "accomplished" "accomplishment" "accomplishments" "accurate" "accurately"

[21] "achievable" "achievement" "achievements" "achievible" "acumen"

[26] "adaptable" "adaptive" "adequate" "adjustable" "admirable"

[31] "admirably" "admiration" "admire" "admirer" "admiring"

[36] "admiringly" "adorable" "adore" "adored" "adorer"

[41] "adoring" "adoringly" "adroit" "adroitly" "adulate"

[46] "adulation" "adulatory" "advanced" "advantage" "advantageous"

[51] "advantageously" "advantages" "adventuresome" "adventurous" "advocate"

[56] "advocated" "advocates" "affability" "affable" "affably"

[61] "affectation" "affection" "affectionate" "affinity" "affirm"

[66] "affirmation" "affirmative" "affluence" "affluent" "afford"

[71] "affordable" "affordably" "afordable" "agile" "agilely"

[76] "agility" "agreeable" "agreeableness" "agreeably" "all-around"

[81] "alluring" "alluringly" "altruistic" "altruistically" "amaze"

[86] "amazed" "amazement" "amazes" "amazing" "amazingly"

[91] "ambitious" "ambitiously" "ameliorate" "amenable" "amenity"

[96] "amiability" "amiabily" "amiable" "amicability" "amicable"

neg[1:100]

[1] "2-faced" "2-faces" "abnormal" "abolish" "abominable"

[6] "abominably" "abominate" "abomination" "abort" "aborted"

[11] "aborts" "abrade" "abrasive" "abrupt" "abruptly"

[16] "abscond" "absence" "absent-minded" "absentee" "absurd"

[21] "absurdity" "absurdly" "absurdness" "abuse" "abused"

[26] "abuses" "abusive" "abysmal" "abysmally" "abyss"

[31] "accidental" "accost" "accursed" "accusation" "accusations"

[36] "accuse" "accuses" "accusing" "accusingly" "acerbate"

[41] "acerbic" "acerbically" "ache" "ached" "aches"

[46] "achey" "aching" "acrid" "acridly" "acridness"

[51] "acrimonious" "acrimoniously" "acrimony" "adamant" "adamantly"

[56] "addict" "addicted" "addicting" "addicts" "admonish"

[61] "admonisher" "admonishingly" "admonishment" "admonition" "adulterate"

[66] "adulterated" "adulteration" "adulterier" "adversarial" "adversary"

[71] "adverse" "adversity" "afflict" "affliction" "afflictive"

[76] "affront" "afraid" "aggravate" "aggravating" "aggravation"

[81] "aggression" "aggressive" "aggressiveness" "aggressor" "aggrieve"

[86] "aggrieved" "aggrivation" "aghast" "agonies" "agonize"

[91] "agonizing" "agonizingly" "agony" "aground" "ail"

[96] "ailing" "ailment" "aimless" "alarm" "alarmed"

Проводим стемминг словарей и удаляем дубликаты

pos <- stemDocument(pos, language = "english")

pos <- pos[!duplicated(pos)]

neg <- stemDocument(neg, language = "english")

neg <- neg[!duplicated(neg)]

pos[1:100]

[1] "a+" "abound" "abund" "access" "acclaim" "acclam"

[7] "accolad" "accommod" "accomod" "accomplish" "accur" "achiev"

[13] "acumen" "adapt" "adequ" "adjust" "admir" "ador"

[19] "adroit" "adul" "adulatori" "advanc" "advantag" "adventuresom"

[25] "adventur" "advoc" "affabl" "affect" "affection" "affin"

[31] "affirm" "affluenc" "affluent" "afford" "aford" "agil"

[37] "agreeabl" "all-around" "allur" "altruist" "amaz" "ambiti"

[43] "amelior" "amen" "amiabl" "amiabili" "amic" "amiti"

[49] "ampl" "ampli" "amus" "angel" "apotheosi" "appeal"

[55] "applaud" "appreci" "appropri" "approv" "ardent" "ardor"

[61] "articul" "aspir" "assur" "astonish" "astound" "astut"

[67] "attent" "attract" "attun" "audibl" "auspici" "authent"

[73] "authorit" "autonom" "avail" "aver" "avid" "award"

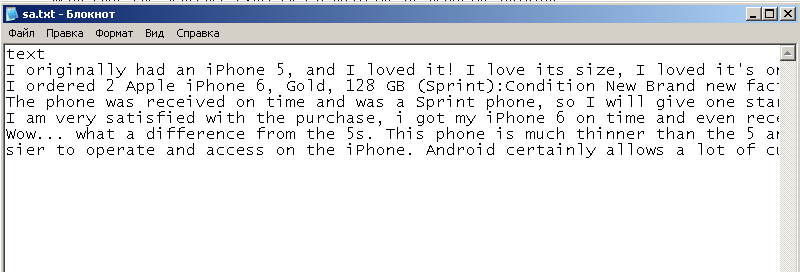
[79] "awe" "awesom" "awestruck" "awsom" "backbon" "balanc"

[85] "bargain" "beauteous" "beauti" "beautifullli" "beautifi" "beckon"

[91] "believ" "belov" "benefactor" "benefic" "benefici" "beneficiari"

[97] "benefit" "benevol" "benifit" "best"

Читаем данные из файла



data<-read.table("sa.txt", header=TRUE,sep=";", encoding="UTF-8", quote="")

# Создаем корпус

reviews <- Corpus(VectorSource(data$text))

reviews[[1]]$content

[1] "I originally had an iPhone 5, and I loved it! I love its size, I loved it's onductivity, but as soon as I came into contact with the iPhone 6, the game was over. I was worried hat it might be a little too big for my taste, but the more I hold it, the more I love it. The responsiveness of the screen, the reaction time of the fingerprint scanner, everything works like a dream. Also, after making multiple phone calls, I have noticed that there's a perfect clarity and that no one has trouble hearing me. You will not be sorry you picked this phone. "

# Выполняем преобразования

reviews <- tm\_map(reviews, content\_transformer(removeNumbers))

reviews[[1]]$content

[1] "I originally had an iPhone , and I loved it! I love its size, I loved it's onductivity, but as soon as I came into contact with the iPhone , the game was over. I was worried hat it might be a little too big for my taste, but the more I hold it, the more I love it. The responsiveness of the screen, the reaction time of the fingerprint scanner, everything works like a dream. Also, after making multiple phone calls, I have noticed that there's a perfect clarity and that no one has trouble hearing me. You will not be sorry you picked this phone. "

reviews <- tm\_map(reviews, content\_transformer(str\_replace\_all), pattern = "[[:punct:]]", replacement = " ")

reviews[[1]]$content

[1] "I originally had an iPhone and I loved it I love its size I loved it s onductivity but as soon as I came into contact with the iPhone the game was over I was worried hat it might be a little too big for my taste but the more I hold it the more I love it The responsiveness of the screen the reaction time of the fingerprint scanner everything works like a dream Also after making multiple phone calls I have noticed that there s a perfect clarity and that no one has trouble hearing me You will not be sorry you picked this phone "

reviews <- tm\_map(reviews, content\_transformer(tolower))

reviews[[1]]$content

[1] "i originally had an iphone and i loved it i love its size i loved it s onductivity but as soon as i came into contact with the iphone the game was over i was worried hat it might be a little too big for my taste but the more i hold it the more i love it the responsiveness of the screen the reaction time of the fingerprint scanner everything works like a dream also after making multiple phone calls i have noticed that there s a perfect clarity and that no one has trouble hearing me you will not be sorry you picked this phone "

reviews <- tm\_map(reviews, content\_transformer(removeWords), words = stopwords("en"))

reviews[[1]]$content

[1] " originally iphone loved love size loved s onductivity soon came contact iphone game worried hat might little big taste hold love responsiveness screen reaction time fingerprint scanner everything works like dream also making multiple ph one calls noticed s perfect clarity one trouble hearing will sorry picked phone "

reviews <- tm\_map(reviews, stemDocument, language = "english")

reviews[[1]]$content

[1] " origin iphon love love size love s onduct soon came contact iphon game worri hat might littl big tast hold love respons screen reaction time fingerprint scanner everyth work like dream also make multipl phone call notic s perfect clariti one troubl hear will sorri pick phone "

reviews

# Создаем матрицу терминов-документов

tdm <- TermDocumentMatrix(reviews, control = list(weighting = weightBin))

# Подсчитываем количество положительных и отрицательных терминов и разность этих значений

pos.mat <- tdm[rownames(tdm) %in% pos, ]

neg.mat <- tdm[rownames(tdm) %in% neg, ]

pos.out <- apply(pos.mat, 2, sum)

neg.out <- apply(neg.mat, 2, sum)

senti.diff <- pos.out - neg.out

senti.diff[senti.diff == 0] <- NA

senti.diff