СОДЕРЖАНИЕ

ВВЕДЕНИЕ..............................................................................................................3

1. Постановка задачи...............................................................................................5

2. Обзор и выбор инструментов для анализа данных..........................................6

3. Описание алгоритма анализа и визуализации данных....................................9

4. Реализация проекта...........................................................................................11

ЗАКЛЮЧЕНИЕ.....................................................................................................25

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ...................................................26

ПРИЛОЖЕНИE А Листинг программы..............................................................27

# **ВВЕДЕНИЕ**

Одним из основных инструментов познания мира является обработка интеллектуальный анализ данных, получаемых человеком из различных источников. Сейчас существует множество методов интеллектуального анализа, и в числе ведущих – методы математической статистики.

В последнее время методы анализа и визуализации данных сильно изменились. С появлением ЭВМ и Интернета объёмы данных увеличились. Коммерческие и государственные организации обладают огромнейшими хранилищами данных. И соответственно, с увеличением объёма данных, представление этой информации в легкодоступном виде становится более сложн~~ой~~ым.

Twitter является популярной социальной сетью, где пользователи могут обмениваться короткими SMS-сообщениями, так называемые твитами. Пользователи делятся мыслями, ссылками и фотографиями в сети Twitter, журналисты комментируют "живые" события, компании продвигают продукты и взаимодействуют с клиентами. Список различных способов использования Twitter может быть очень большим, с 500 миллионами твитов в день, есть много данных для анализа и визуализации.

Разработчики социальной сети Twitter предоставляют открытый API (интерфейс программирования приложений), который разрешает извлекать информацию о сообщениях из базы данных твиттера по любой тематике.

Главная задача анализа и визуализации — отобразить главную информацию о свойствах системы или явления максимально легким для восприятия способом. В идеальном случае анализ системы и визуализацию его результатов можно сделать в рамках одного инструмента. Например, R с его обширным набором пакетов позволяет это.

В данном курсовом проекте поставлена задача анализа и визуализации данных из социальной сети Twitter, которая позволила бы пользователю наглядно отследить статистику тех или иных популярных тематик в последнее время.

# **1.Постановка задачи**

Цель**:** Освоить средства и методы сбора и визуализации данных из социальной сети Twitter с помощью языка R.

Необходимо разработать программный код на языке R, который позволит собирать наиболее популярные ключевые слова, которые соответствуют определённому #хэштэгу с сайта twitter.com в определённый период времени.

Проанализировав поставленную задачу, я пришёл к выводу, что для лучшей читабельности и поддержки целесообразно написать две функции.

Первая функция - извлечение статистических данных из Twitter:

GetStat (Значение1),

где Значение1 - является #хэштэгом, который содержит любую строку.

Вторая функция - визуализация данных:

SetView(Значение2)

где Значение2 - это список, который содержит твит в следующем виде:

"iokek: Только что пришёл с пробежки. Очень устал! #sport #Russia"

2. Обзор и выбор инструментов для анализа данных

Перед началом выполнения поставленной задачи были проанализированы зарубежные и отечественные источники на тему: «Популярность языков анализа данных». В соответствии с веб-сайтом американской компании по бизнес-аналитике проводились голосования на выбор программного обеспечения для анализа данных в 2012, 2013, 2014 годах. Топ-3 голосов составили языки R, SAS, Python. И на первом месте лидировал язык R. Дело в том, что в отличие от своих конкурентов язык R имеет более профильную специализацию, так как его используют в основном для статистической обработки данных и построения графиков. Это делает язык R основным инструментом, который применяется специалистами по анализу данных, и одновременно ограничивает сферу применения.

Таблица 1. Сравнительная характеристика средств анализа данных [8].

|  |  |
| --- | --- |
| **What programming/statistics languages you used for an analytics / data mining / data science work in 2014?** | |
| **Language used** | http://www.kdnuggets.com/polls/graph/g0.gif % voters in 2014 (719 total)  http://www.kdnuggets.com/polls/graph/g4.gif % voters in 2013 (713 total)  http://www.kdnuggets.com/polls/graph/g8.gif % voters in 2012 (579 total) |
| R (352 voters in 2014) | http://www.kdnuggets.com/polls/graph/g0.gif 49.0% http://www.kdnuggets.com/polls/graph/g4.gif 60.9% http://www.kdnuggets.com/polls/graph/g8.gif 52.5% |
| SAS (262) | http://www.kdnuggets.com/polls/graph/g0.gif 36.4% http://www.kdnuggets.com/polls/graph/g4.gif 20.8% http://www.kdnuggets.com/polls/graph/g8.gif 19.7% |
| Python (252) | http://www.kdnuggets.com/polls/graph/g0.gif 35.0% http://www.kdnuggets.com/polls/graph/g4.gif 38.8% http://www.kdnuggets.com/polls/graph/g8.gif 36.1% |
| SQL (220) | http://www.kdnuggets.com/polls/graph/g0.gif 30.6% http://www.kdnuggets.com/polls/graph/g4.gif 36.6% http://www.kdnuggets.com/polls/graph/g8.gif 32.1% |
| Java (89) | http://www.kdnuggets.com/polls/graph/g0.gif 12.4% http://www.kdnuggets.com/polls/graph/g4.gif 16.5% http://www.kdnuggets.com/polls/graph/g8.gif 21.2% |

Продолжение таблицы 1.

|  |  |
| --- | --- |
| Unix shell/awk/sed (63) | http://www.kdnuggets.com/polls/graph/g0.gif 8.8% http://www.kdnuggets.com/polls/graph/g4.gif 11.1% http://www.kdnuggets.com/polls/graph/g8.gif 14.7% |
| Pig Latin/ Hive/ other Hadoop-based languages (61) | http://www.kdnuggets.com/polls/graph/g0.gif 8.5% http://www.kdnuggets.com/polls/graph/g4.gif 8.0% http://www.kdnuggets.com/polls/graph/g8.gif 6.7% |
| SPSS (58) | http://www.kdnuggets.com/polls/graph/g0.gif 8.1% not asked not asked |
| MATLAB (45) | http://www.kdnuggets.com/polls/graph/g0.gif 6.3% http://www.kdnuggets.com/polls/graph/g4.gif 12.5% http://www.kdnuggets.com/polls/graph/g8.gif 13.1% |
| Scala (28) | http://www.kdnuggets.com/polls/graph/g0.gif 3.9% http://www.kdnuggets.com/polls/graph/g4.gif 2.2% http://www.kdnuggets.com/polls/graph/g8.gif 2.4% |
| C/C++ (26) | http://www.kdnuggets.com/polls/graph/g0.gif 3.6% http://www.kdnuggets.com/polls/graph/g4.gif 9.3% http://www.kdnuggets.com/polls/graph/g8.gif 14.3% |

Соответственно для анализа и обработки данных был выбран язык программирования R. R – это и язык программирования, и среда для статистических вычислений и графического анализа, который был первоначально разработан в Bell Laboratories, свободно распространяется, поддерживается большим и активным исследовательским сообществом по всему миру. Как было указано выше, помимо R существует много популярных инструментов для статистической и графической обработки данных, однако R имеет много достоинств [3]:

* многие статистические программы стоят больших денег, тогда как среда R полностью бесплатна. Если вы студент, то выбор очевиден;
* R – это мощная статистическая программа, в которой реализованы все способы анализа данных;
* R имеет самые современные графические возможности. Если требуется визуализировать сложные данные, то в R реализованы самые разнообразные и мощные методы анализа данных из доступных;
* получение данных из разных источников. R может импортировать данные из веб-сайтов, текстовых файлов, Excel файлов, систем управления базами данных, и других хранилищ данных. R может также экспортировать данные в форматах всех этих систем;
* R не имеет аналогов для простого написания программ, которые реализуют новые статистические методы;
* R работает на многих операционных системах, включая Windows, Unix и Mac OS X;
* для R написано огромное количество полезных функций, реализующих известные методы анализа данных.

3. Описание алгоритма анализа и визуализации данных

Начало

Импорт данных

Подготовка данных, знакомство с ними, исправление ошибок

Подбор статистической модели

Оценка модели на новых данных

Результат

Конец

Рисунок 1 - Блок-схема процесса обработки данных

Теперь рассмотрим по пунктам:

* Импорт данных. Чтобы работать с данными, нам нужно их сначала получить. Данные считываются с веб-страниц в векторы. А затем заносятся в Excel таблицу или обрабатываются в R.
* Подготовка данных, знакомство с данными. Исправление ошибок. Здесь нужно ознакомиться с данными, правильно ли мы их считали и так далее.
* Подбор статистической модели. Если данные считаны правильно, то приступаем к подбору нужных нам моделей для визуализации данных.
* Оценка модели на новых данных. В этом пункте мы смотрим на адекватность нашей модели, а именно, полностью ли мы исследовали данные и ответили на все относящиеся к делу вопросы, на которые можно было ответить.
* Результат. Результат интерпретации программного кода.

4. Реализация проекта

Для сбора твиттер-сообщений будет использована библиотека twitteR, а для визуализации будет использована библиотека wordcloud.

Перед тем как получить данные с twitter.com необходимо получить доступ к своему API. Подключаем нужные библиотеки.

library(twitteR)

Наиболее используемой функцией из пакета twitteR является searchTwitter(), аргументом которой является текстовый вектор, определяющий поисковую фразу. Для реализации проекта было решено взять хэштэг #Russia.

searchTerm="#Russia"

Начинаем поиск и сохраняем результаты в объекте temptweets. В качестве даты публикации твитов можно указать текущую дату (команда Sys.Date()) или любую дату.

temptweets <- searchTwitter(searchTerm, n = 1500, lang="ru", since = as.character(Sys.Date()))

Выводим размер полученного объекта:

length(temptweets)

После того, как получили данные, необходимо их подготовить для дальнейшего анализа.

Преобразуем список твитов temptweets в таблицу данных функцией twListToDF() из пакета twitteR:

tw.df <- twListToDF(temptweets)

В некоторых из полученных сообщений часто будут встречаться имена пользователей, которые начинаются со знака @. Информация такого рода не представляет никакого интереса и является "шумовой". Чтобы избавиться от имён пользователей, используем регулярные выражения:

# Удаляем пользовательские имена:

tweets <- gsub("@\\w+", "", temptweets)

Далее подключим библиотеку tm (от text mining), которая имеет обширные возможности для анализа текстовой информации. Обработка твитов, хранящихся в текстовом векторе tweets, потребует нескольких функций из этого пакета. Первым шагом должно стать создание "корпуса", т.е. сведение всех имеющихся небольших текстов (твитов) в некоторый один объект:

tw.corpus <- Corpus(VectorSource(tweets))

Теперь с корпусом текстов можно исполнять различные действия с помощью функции tm\_map() из библиотеки tm. Для начала удаляем из собранных twitter-сообщений все знаки пунктуации, с помощью функции removePunctuation():

tw.corpus <- tm\_map(tw.corpus, removePunctuation)

Теперь сделаем так, чтобы все слова в twitter-сообщениях были представлены прописными буквами:

tw.corpus <- tm\_map(tw.corpus, tolower)

Следующим шагом необходимо удалить из twitter-сообщений как можно больше стоп-слов, т.е слов, которые не несут никакого смысла (например: им, я, в, перед, это и так далее). Для этого используется функция removeWords() из той же библиотеки tm. К сожалению, для русскоязычных текстов нет встроенной функции для работы со стоп-словами. Поэтому был создан список стоп-слов вручную. Выполняем следующую команду:

tw.corpus <- tm\_map(tw.corpus, removeWords, my.stopwords)

где my.stopwords - список стоп-слов.

Далее выполним визуальный анализ облака наиболее часто встречающихся в этих сообщениях слов. Для создания облака слов воспользуемся возможностями библиотеки wordcloud. Также потребуется библиотека RColorBrewer для создания палитры цветов.

generateWordcloud <- function(corpus, min.freq = 3, ...) {

require(wordcloud)

require(RColorBrewer)

Преобразовываем "корпус" в матрицу индексируемых слов из сообщений.

doc.m <- TermDocumentMatrix(corpus, control = list(minWordLength = 1))

dm <- as.matrix(doc.m)

Подсчитываем частоту слов:

v <- sort(rowSums(dm), decreasing = TRUE)

d <- data.frame(word = names(v), freq = v)

pal <- brewer.pal(8, "Accent")

Генерируем облако слов:

wc <- wordcloud(d$word, d$freq, min.freq = min.freq, colors = pal)

wc

}

Выводим облако часто встречающихся слов:

print(generateWordcloud(my.corpus), 15)



ЗАКЛЮЧЕНИЕ

Итак, анализ и визуализация данных необходима всегда, когда результат неочевиден, и часто даже тогда, когда он кажется очевидным.

Можно проводить сравнения между разными данными. Например, можно выяснить, популярность тех или иных тематик, больше или меньше комментариев при помощи диаграмм и даже графов, т.е какая тема вызывает больше интересов в социальной сети. Кроме того, модератор сайта может отфильтровывать потенциально не интересные темы.

В ходе исследования курсового проекта я узнал, как извлекать данные из социальной сети Twitter. Научился подготавливать данные для дальнейшего исследования. Рассмотрел несколько способов визуализации на основе этих данных. И выполнил их обработку и анализ с помощью средств языка R.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. А.Б.Шипунов. Е.М. Балдин. П.А. Волкова. [и др.] Наглядная статистика. Используем R! // М.: ДМК-Пресс. 2014.— C. 293.
2. Мастицкий С.Э., Шитков В.К. Статистический анализ и визуализация данных с помощью R. // М.: ДМК. 2014.— C.188.
3. Роберт И. Кабаков. R в действии. Анализ и визуализация данных в программе R / пер. с англ. Полины А. Волковой. // М.: ДМК Пресс, 2014. – C. 588.
4. Общая документация [Электронный ресурс] Статья, URL: http://cran.r-project.org/ [Дата обращения: 5 ноября 2015].
5. Визуализация статических и динамических сетей на R. [Электронный ресурс] Статья, URL:http://[habrahabr.ru/company/infopulse/blog/262079](file:///C:\Users\Алексей\AppData\Roaming\Microsoft\Word\www.habrahabr.ru\company\infopulse\blog\262079) [Дата обращения: 26 ноября 2015].
6. R с нуля [Электронный ресурс] Статья, URL:http://[kstera.ru/R/index.htm](file:///C:\Users\Алексей\AppData\Roaming\Microsoft\Word\www.kstera.ru\R\index.html)l [Дата обращения: 5 ноября 2015].
7. How to Webscrape in R, the Rvest and pipeR way. [Электронный ресурс] Статья, URL: http://[asbcllc.com/blog/2014/november/creating\_bref\_scraper](http://asbcllc.com/blog/2014/november/creating_bref_scraper/) [Дата обращения: 15 ноября 2015].
8. Four main languages for Analytics, Data Mining, Data Science. [Электронный ресурс] Статья, URL:http://[kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html](http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html) [Дата обращения: 1 декабря 2015].

ПРИЛОЖЕНИЕ А

Листинг программы

library(tm)

library(wordcloud)

library(twitteR)

library(RCurl)

consumer\_key="Q2yPlpVRwIySuv51LNM4hnPUB"

consumer\_secret="MVMIygXfikFeJ9UQuhbFZJPlDLjBWV6Q2WUpaJab6fUxf6vnAT"

access\_token="2303992373-12wcsJMgHkzUdgU4bydwHIeo2QqP0PaQEt8GQrM"

access\_secret="tHtg7kojSqGrVVUBuf52vVwVAxXgoZiO20fDy71OHUHqq"

setup\_twitter\_oauth(consumer\_key,consumer\_secret,access\_token, access\_secret)

searchTerm="#sport"

temptweets <- searchTwitter(searchTerm, n = 100, lang="ru", since = '2016-05-09' )

length(temptweets)

tw.df <- twListToDF(temptweets)

RemoveAtPeople <- function(tweet) {

gsub("@\\w+", "", tweet)}

tweets <- as.vector(sapply(tw.df$text, RemoveAtPeople))

my.stopwords <- c("в", "без", "до", "из", "к", "на", "по", "о", "от", "перед", "при", "через", "за", "над", "об", "под", "про", "для", "вблизи", "вглубь", "вдоль", "возле", "около", "вокруг", "впереди", "после", "посредством", "в роли", "в зависимости от", "путём", "насчёт", "по поводу", "ввиду", "по случаю", "в течение", "благодаря", "несмотря на", "спустя", "с ", "из-под", "из-за", "по-над", "в отличие от", "в связи", "как", "словно", "так как", "для того чтобы", "тоже", "зато", "чтобы", "также", "потому что", "и ", "а ", "что", "или", "но", "однако", "когда", "лишь", "едва", "где", "куда", "откуда", "столько", "настолько", "так", "до такой степени", "до того", "такой", "как будто", "будто", "точно", "как бы","если", "если бы", "коли", "ежели", "несмотря на то", "хотя", "хоть", "пускай", "дабы", "с тем чтобы", "так что", "ли", "не", "какой")

generateCorpus <- function(df, my.stopwords = c()) {

tw.corpus <- Corpus(VectorSource(df))

tw.corpus <- tm\_map(tw.corpus, removePunctuation)

tw.corpus <- tm\_map(tw.corpus, content\_transformer(tolower))

tw.corpus <- tm\_map(tw.corpus, removeWords, my.stopwords)

tw.corpus

}

generateWordcloud <- function(corpus, min.freq = 3, ...) {

require(wordcloud)

doc.m <- TermDocumentMatrix(corpus,

control = list(minWordLength = 1))

dm <- as.matrix(doc.m)

v <- sort(rowSums(dm), decreasing = TRUE)

d <- data.frame(word = names(v), freq = v)

require(RColorBrewer)

pal <- brewer.pal(8, "Dark2")

wc <- wordcloud(d$word, d$freq, min.freq = min.freq, colors = pal)

wc

}

print(generateWordcloud(my.corpus), 15)