**Лабораторная работа №7. обработка естественного языка на R**

**Цель работы**: приобрести основные навыки работы с языком  
программирования R, методами работы с текстом и со строками

**Установка необходимых библиотек**

* NLP [1]
* openNLP [2]
* RWake [3]
* qdap [4]

install.packages(c("NLP", "openNLP", "RWeka", "qdap"))

Предположим, успешно установили все библиотеки, что нам нужно, чтобы загрузить следующие библиотеки.

library(NLP)

library(openNLP)

library(RWeka)

**Токенизация**

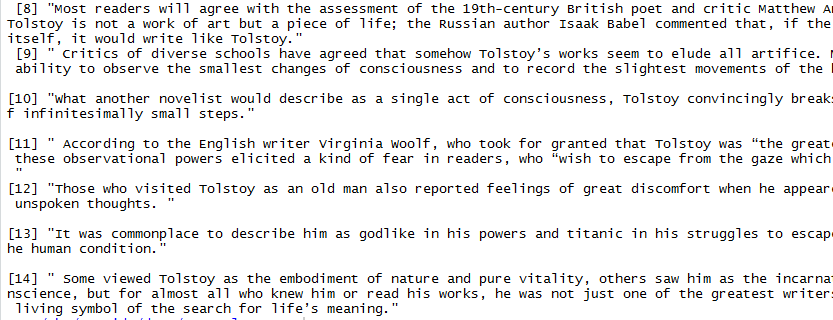
В данном примере, использовали текстовый файл биографии автора Толстого Л.Н. [5]. Мы научимся читать текстовый файл в R, разбивать его на слова и предложения и превращать его в n-граммы. Все они называются токенизацией, потому что мы разбиваем текст на единицы значения, называемые токенами.

readfile <- file("F:/PhD/R\_guide/data/Leo\_Tolstoy.txt",open = "r")

bio <- readLines(readfile)

print(bio)

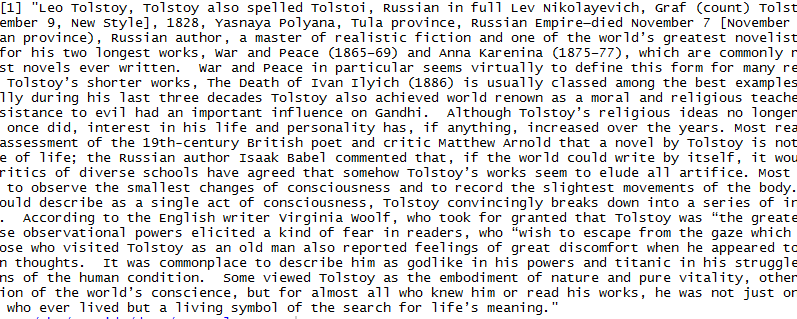
close(readfile)



Видим, что в файле содержится четырнадцать строк, каждый из которых содержится в символьном векторе. Мы можем объединить все эти векторы символов в один вектор символов, используя функцию paste (), добавляя пробел между каждым из них.

bio <- paste(bio, collapse = " ")

print(bio)



**Предложение и текстовые аннотации**

Сначала мы загружаем необходимые библиотеки

library(NLP)  
library(openNLP)  
library(magrittr)

Для NLP мы обязаны использовать класс String. Нам нужно преобразовать нашу переменную bio в строку.  
  
bio <- as.String(bio)

Затем нам нужно создать аннотаторы для слов и предложений. Аннотаторы создаются функциями, которые загружают базовые библиотеки Java. Эти функции затем отмечают места в строке, где слова и предложения начинаются и заканчиваются. Функции аннотации сами по себе создаются функциями.  
  
word\_ann <- Maxent\_Word\_Token\_Annotator()  
sent\_ann <- Maxent\_Sent\_Token\_Annotator()

Эти аннотаторы образуют “pipeline” для аннотирования текста в нашей переменной bio. Сначала мы должны определить где предложения, тогда мы можем определить где слова. Мы можем применить эти функции аннотатора к нашим данным с помощью функции annotate ().

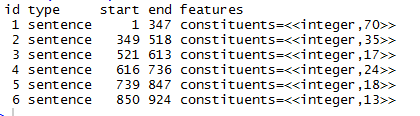
bio\_ann <- annotate(bio, list(sent\_ann, word\_ann))

Результатом является объект аннотации. Рассматривая несколько первых элементов, содержащихся в объекте, мы можем получить информацию, содержащуюся в объекте аннотаций.

class(bio\_ann)



head(bio\_ann)



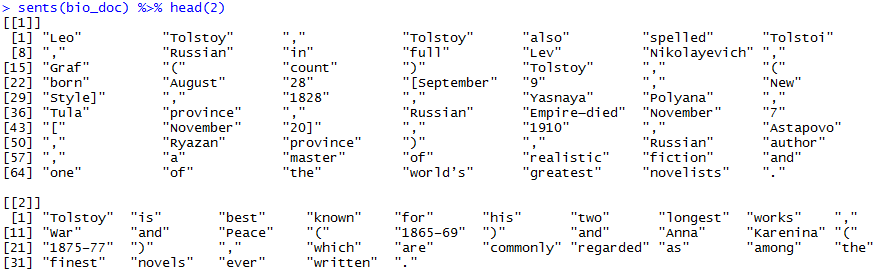
Мы видим, что объект аннотации содержит список предложений (а также слов), идентифицированных положением. То есть первое предложение в файле начинается с символа 1 и заканчивается символом 348. Предложения также содержат информацию о положениях слов, которые их содержат.

Мы можем объединить биографию и аннотации, чтобы создать то, что пакет NLP вызывает AnnotatedPlainTextDocument.

bio\_doc <- AnnotatedPlainTextDocument(bio, bio\_ann)

Теперь мы можем извлечь информацию из нашего документа, используя функции доступа, такие как sents(), чтобы получить предложения и words(), чтобы получить слова. Мы могли бы получить только обычный текст с as.character (bio\_doc).

sents(bio\_doc) %>% head(2)



words(bio\_doc) %>% head(10)



Это уже полезно, поскольку мы могли бы использовать результирующие списки слов/предложений для выполнения других видов вычислений. Но есть и другие виды аннотаций, которые более непосредственно относятся к историкам.

Задачи

1. Реализовать все примеры указные в лабораторной работе на своих  
   произвольных текстах.