

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ  
Кафедра ИТАС  
Факультет Информационных Технологий и Управления

## **Технология анализа текста и извлечения ключевых слов**

Выполнил: ст. гр. 820601 Шведов А.Р.

Руководитель: Ярмолик В.И.

Минск, 2020

Целью работы является практическое освоение технологии анализа текста, извлечения ключевых слов и профессионального поиска информации.

- Общие принципы функционирования поисковых средств.
- Правильный выбор ключевых слов поиска.

Джордж Зипф установил, что все тексты подчиняются общим закономерностям, и сформулировал в 1946—49 гг. несколько законов, которые нашли применение в технологии поиска информации.

## 1-й Закон Зипфа «Ранг - частота»

*Частотой встречаемости слова* называется величина, равная числу вхождений слова в текст. Вероятность обнаружения некоторого слова в тексте  $P$  равна отношению частоты его вхождения к общему числу слов в тексте.

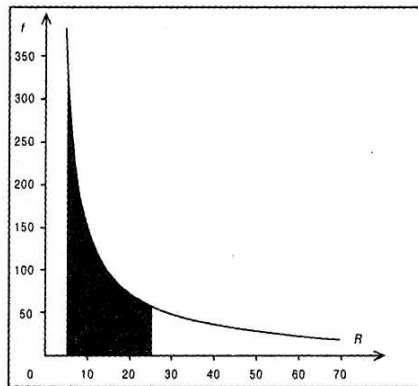
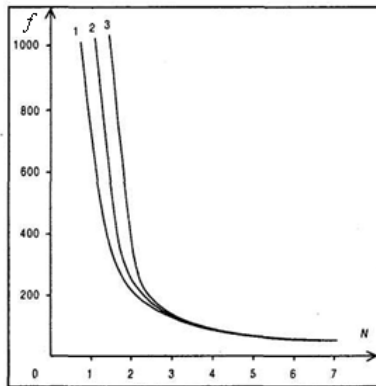


Рисунок 1: Кривая зависимости частоты встречаемости слова от его ранга.

### 2-й Закон Зипфа «количество - частота»

Частота и количество разных слов  $N$ , входящих в текст с данной частотой, также связаны между собой определенной зависимостью. Если построить график, отложив по оси ординат частоту вхождения слова, а по оси абсцисс — количество разных слов, характеризуемых одинаковой частотой, то получившаяся кривая будет сохранять свои параметры для всех без исключения созданных человеком текстов в пределах одного языка.



**Рисунок 2:** Кривые для французского (кривая 1), английского (кривая 2) и русского (кривая 3) языков.

Возможность извлечения ключевых слов из текстовых материалов имеется и в текстовом редакторе *MS Word*, однако использование этой возможности дает неудовлетворительные результаты. В данной работе будут использоваться утилиты командной строки *bash*, такие как *find*, *grep*, *awk*, *tr*.



### Bash

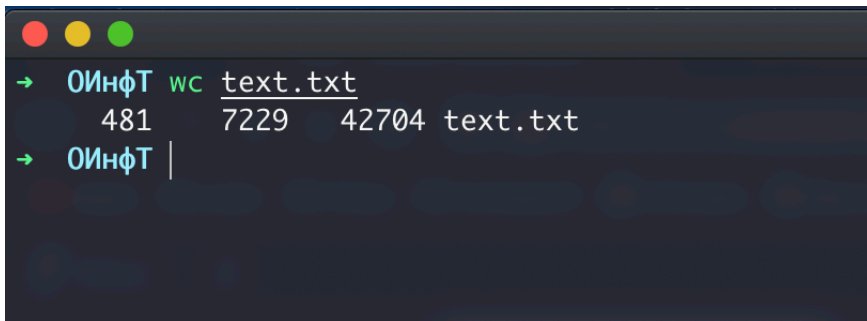
*Bash* - это одна из самых известных командных оболочек Linux. Она позволяет выполнять различные команды ОС, а также наборы команд, оформленные в виде файлов, так называемые скрипты или сценарии. С его помощью можно реализовывать конструкции циклов и ветвлений, перенаправлять ввод-вывод в файлы, считывать параметры из файлов, с клавиатуры, использовать переменные и т.д.

### AWK

*Awk* – утилита предназначенная для простых, механических и вычислительных манипуляций над данными. Утилита изначально объединяла свойства утилит *UNIX* - *sed* и *grep*. В дальнейшем ее возможности значительно расширились. По-умолчанию, поля – это последовательности символов, отделенные друг от друга пробелами, однако имеется возможность назначения других символов, в качестве разделителя полей. Она анализирует и обрабатывает каждое поле в отдельности.

### **tr**

Утилита *tr* выполняет преобразование, подстановку (замену), сокращение и/или удаление символов, поступающих со стандартного ввода, записывая результат на стандартное устройство вывода. Она часто применяется для удаления управляющих символов из файла или преобразования регистра символов.



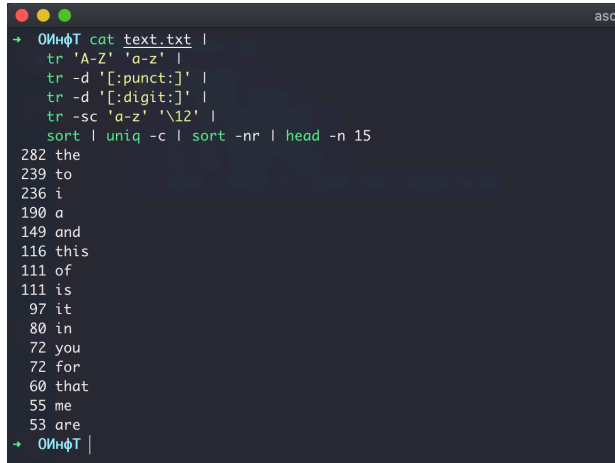
```
→ ОИнфТ wc text.txt
    481    7229   42704 text.txt
→ ОИнфТ |
```

Рисунок 3: Количество слов и строк в файле.



```
→ ОИДФТ tr ' ' '\n' < text.txt | grep "this" | wc -l
85
→ ОИДФТ tr ' ' '\n' < text.txt | grep "code" | wc -l
46
→ ОИДФТ tr ' ' '\n' < text.txt | grep "are" | wc -l
76
→ ОИДФТ |
```

Рисунок 4: Подсчет разных слов.

A terminal window with a dark background and light green text. The window has three colored window control buttons (red, yellow, green) in the top-left corner and the text 'as0' in the top-right corner. The terminal shows a series of commands being executed, followed by the output of the final command. The commands are: 'cat text.txt |', 'tr 'A-Z' 'a-z' |', 'tr -d '[:punct:]' |', 'tr -d '[:digit:]' |', 'tr -sc 'a-z' '\12' |', and 'sort | uniq -c | sort -nr | head -n 15'. The output consists of 15 lines, each starting with a count followed by a word: '282 the', '239 to', '236 i', '190 a', '149 and', '116 this', '111 of', '111 is', '97 it', '80 in', '72 you', '72 for', '60 that', '55 me', and '53 are'. The prompt '→ ОИИФТ' is visible at the bottom of the terminal.

```
→ ОИИФТ cat text.txt |  
    tr 'A-Z' 'a-z' |  
    tr -d '[:punct:]' |  
    tr -d '[:digit:]' |  
    tr -sc 'a-z' '\12' |  
    sort | uniq -c | sort -nr | head -n 15  
282 the  
239 to  
236 i  
190 a  
149 and  
116 this  
111 of  
111 is  
97 it  
80 in  
72 you  
72 for  
60 that  
55 me  
53 are  
→ ОИИФТ |
```

Рисунок 5: Полный анализ текста

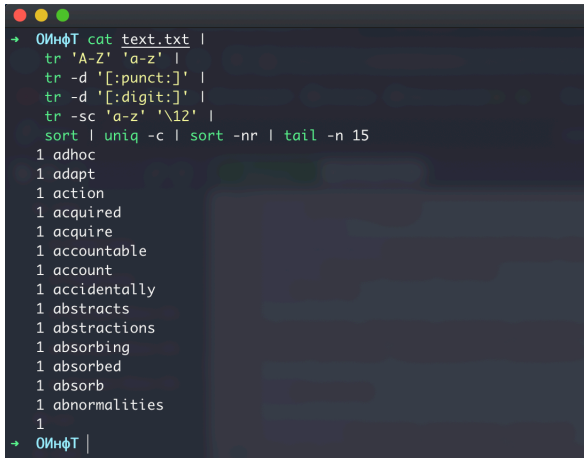
```
1 cat text.txt |  
2   tr 'A-Z' 'a-z' |  
3   tr -d '[:punct:]' |  
4   tr -d '[:digit:]' |  
5   tr -sc 'a-z' '\12' |  
6   sort | uniq -c | sort -nr | head -n 30
```

Листинг 1: Финальный код программы.

### Создание поискового запроса

Так как проанализированный текст является художественно-разговорным, попробуем создать поисковой запрос согласно уникальным словам с малым рангом. Выведем 10 таких слов с помощью программы из п4.3 – Рисунок 6





```
→ ОИИФТ cat text.txt |  
  tr 'A-Z' 'a-z' |  
  tr -d '[:punct:]' |  
  tr -d '[:digit:]' |  
  tr -sc 'a-z' '\12' |  
  sort | uniq -c | sort -nr | tail -n 15  
1  adhoc  
1  adapt  
1  action  
1  acquired  
1  acquire  
1  accountable  
1  account  
1  accidentally  
1  abstracts  
1  abstractions  
1  absorbing  
1  absorbed  
1  absorb  
1  abnormalities  
1  
→ ОИИФТ |
```

Рисунок 6: Уникальные слова

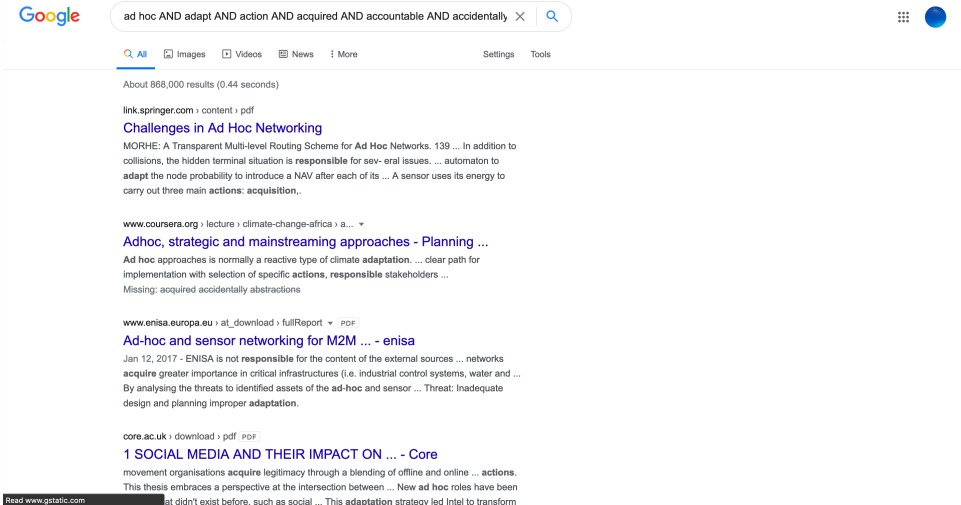


Рисунок 7: Поиск по уникальным словам

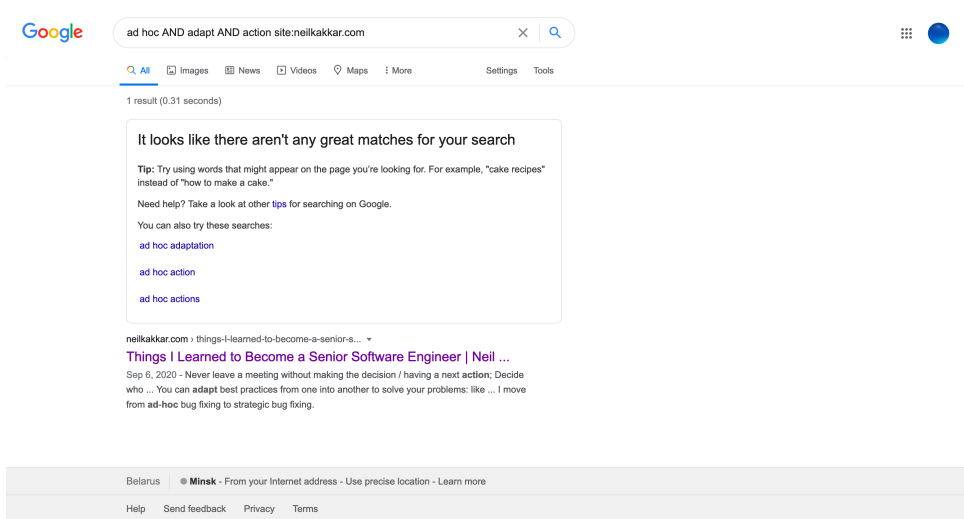


Рисунок 8: Поиск исходного текста.

