СЕТЕВЫЕ ИССЛЕДОВАНИЯ®

Метода анализа текста в R

Алексей Горгадзе
Анастасия Кузнецова

NET-RESEARCH.NET

Чистка данных

```
library(tm); library(tidytext)

text <- gsub("[^[:alnum:]]", " ", text)

text <- gsub("[a-zA-Z0-9]+", "",  text)

text <- tolower(corp)

text <- removeNumbers(text)

text <- removePunctuation(text)

text <- removeWords(text, stopwords("russian"))
```

```
text <- removeWords(text, stoplist)
```

<u>В стоп слова входят (stoplist):</u>

- слишком частотные
- слишком редкие
- слишком короткие
- не существительные
- имена собственные

Лемматизация

Приведение словоформы к лемме (к инфинитиву)

MyStem (Яндекс) - производит морфологический анализ текста на русском языке

text.tmp <- system2("mystem", c("-c", "-l", "-d"), input=docs$text, stdout=TRUE) (должен быть установлен MyStem)

кошками -> кошка

Стемминг (урезание слова до основы):
кошками -> кошк

Форматы текстовых данных

corpus1 <- Corpus(VectorSource(text), readerControl=list(language='ru'))

tdm.matrix <- TermDocumentMatrix(corpus1) / dtm.matrix <- DocumentTermMatrix(corpus1)

words_matrix <- as.matrix(tdm.matrix)

term-document matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ад	1	0	0	0	1	0	0	0	0	0	0	0	0	0	
ангел	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
бояться	1	0	0	0	0	0	0	0	0	0	0	0	2	0	
внутри	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
возникать	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
вопрос	3	0	0	0	0	0	0	0	1	0	0	0	3	0	
впечатление	1	0	0	0	0	0	0	0	0	1	2	0	0	0	
второй	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
вырастать	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
герой	1	0	0	0	0	3	0	2	1	2	1	0	0	0	0
голова	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0
грязь	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Name	Type	Value
wCorpus	list [2392] (S3: SimpleCorpus,	List of length 2392
1	list [2] (S3: PlainTextDocumer	List of length 2
content	character [1]	'какой-то пустота образовываться внутри я после завершение...
meta	list [7] (S3: TextDocumentMet	List of length 7
author	character [0]	
datetimestamp	list [1] (S3: POSIXlt, POSIXt)	List of length 1
[[1]]	double [1]	31.43707
description	character [0]	
heading	character [0]	
id	character [1]	'1'
language	character [1]	'ru'
origin	character [0]	
2	list [2] (S3: PlainTextDocumer	List of length 2
3	list [2] (S3: PlainTextDocumer	List of length 2

Частотность слов

words_freq <- sort(rowSums(words_matrix), decreasing=TRUE)

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

Облака слов

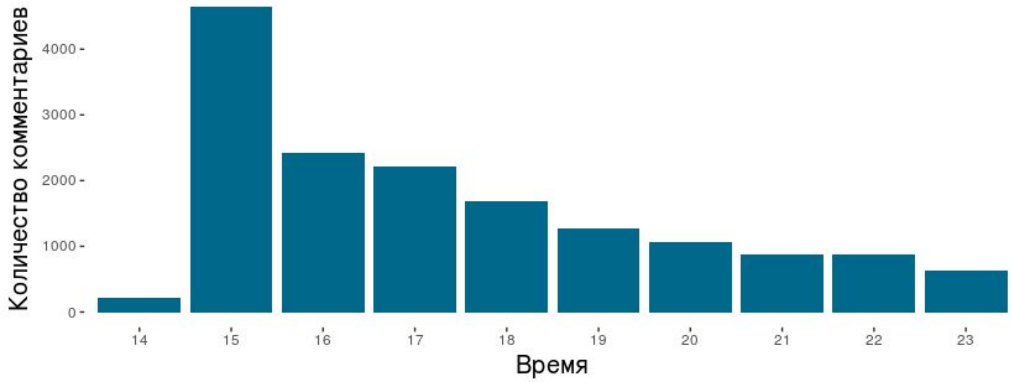

```
wordcloud(words = words_freq$word, freq =
words_freq$freq, scale=c(2,.2), min.freq = 5,
max.words=Inf, random.order=FALSE, rot.per=0.1,
ordered.colors=FALSE,
random.color=TRUE,colors=pal2)
```

14:45 - 15:59

16:00 - 18:59

19:00 - 21:59

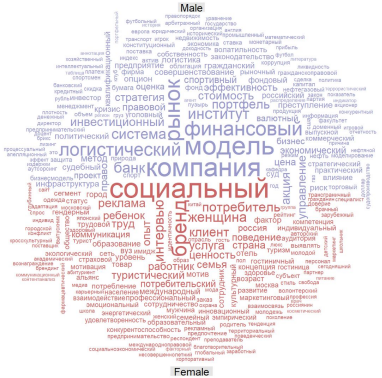
Количество комментариев по часам

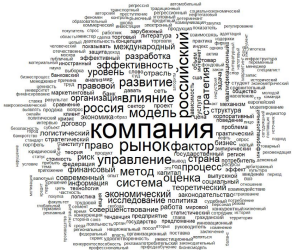


21:59 - 23:37

Сравнение частотности

```
Male <- as.matrix(sort(sapply(dtm1, "sum"), decreasing = T)
            [1:length(dtm1)], colnames = count)
Female <- as.matrix(sort(sapply(dtm2, "sum"), decreasing =
T)
            [1:length(dtm2)], colnames = count)

# Removing missing values
Male <- Male[complete.cases(Male),]
Female <- Female[complete.cases(Female),]

words1 <- data.frame(Male)
words2 <- data.frame(Female)
```

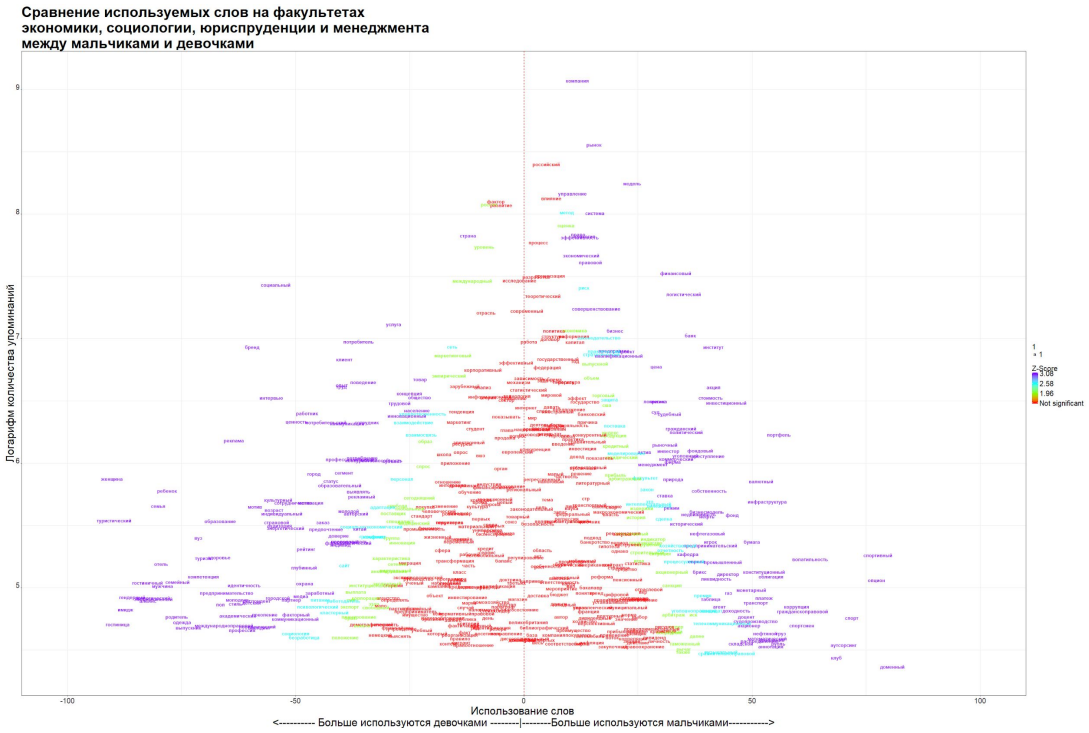




```
# Merge the two tables by row names
wordsCompare <- merge(words1, words2, by="row.names", all = T)
# Replace NA with 0
wordsCompare[is.na(wordsCompare)] <- 0

term.matrix <- as.matrix(wordsCompare[,2:3])
rownames(term.matrix) <- wordsCompare[,1]

png("Name_2wc_300.png", width=2000,height=1200)
par(mfrow=c(1,2),oma = c(2, 1, 5, 1))
comparison.cloud(term.matrix,max.words=300, colors = c("#9999CC",
"#CC6666"), scale=c(8, 1))
title("", sub = "Название левого вордклауда",
      cex.main = 5,   font.main= 2, col.main= "black",
      cex.sub = 3, font.sub = 1, col.sub = "black")
commonality.cloud(term.matrix,random.order=FALSE,max.words=300,
scale=c(8, 1), main="Plot 1")
title("", sub = "Название правого вордклауда",
      cex.main = 5,   font.main= 2, col.main= "black",
      cex.sub = 3, font.sub = 1, col.sub = "black")
mtext("Общее название", outer = TRUE, cex = 5, font = 2)
dev.off()
```

Сравнение частотностей

wordsCompare$prop <- wordsCompare$Male/sum(wordsCompare$Male)
wordsCompare$prop2 <- wordsCompare$Female/sum(wordsCompare$Female)




```
# Broke down the z score formula a little to understand how it worked
a <- wordsCompare$prop
b <- wordsCompare$prop2
c <- wordsCompare$Male
d <- wordsCompare$Female
e <- sum(c)
f <- sum(d)

# z score formula - adds column for z scores
wordsCompare$z <- (a - b) / ((sqrt(((sum(c) * a) + (sum(d) * b)) / (sum(c) +
                  sum(d)) * (1 - ((sum(c) * a) + (sum(d) * b)) / (sum(c) +
                  sum(d))))) * (sqrt((sum(c) + sum(d)) / (sum(c) *
                  sum(d)))))

# calculate percentage reduction:
wordsCompare$dif1 <- -100 * (1 - wordsCompare$prop/wordsCompare$prop2)
# calculate percentage increase
wordsCompare$dif2 <- 100 * (1 - wordsCompare$prop2/wordsCompare$prop)
                  ----------------------------------------------------------
require(ggplot2)

png("eco_dif_m_f_words_full_size_byDif.png", width=3000,height=1500)
ggplot(wordsCompare3, aes(dif, log(abs(Male + Female)), size =
1,label=Row.names, colour = z2))+
      scale_colour_gradientn(name="Z Score", colours=c("#80FF00FF",
"#00FFFFFF", "#8000FFFF")) +
      geom_text(fontface = 2, alpha = .8) +
      #scale_size(range = c(3, 12)) +
      ylab("Логарифм от каоличества упоминаний") +
      xlab("Использование слов \n <----------Больше используются
девушками --------|--------Больше используются мальчиками---------->")+
      geom_vline(xintercept=0, colour  = "red", linetype=2)+
      theme_bw() + #theme(legend.position = "none") +
      ggtitle("Сравнение используемых слов на факультете экономики
\nмежду мальчиками и девушками")
dev.off()
```

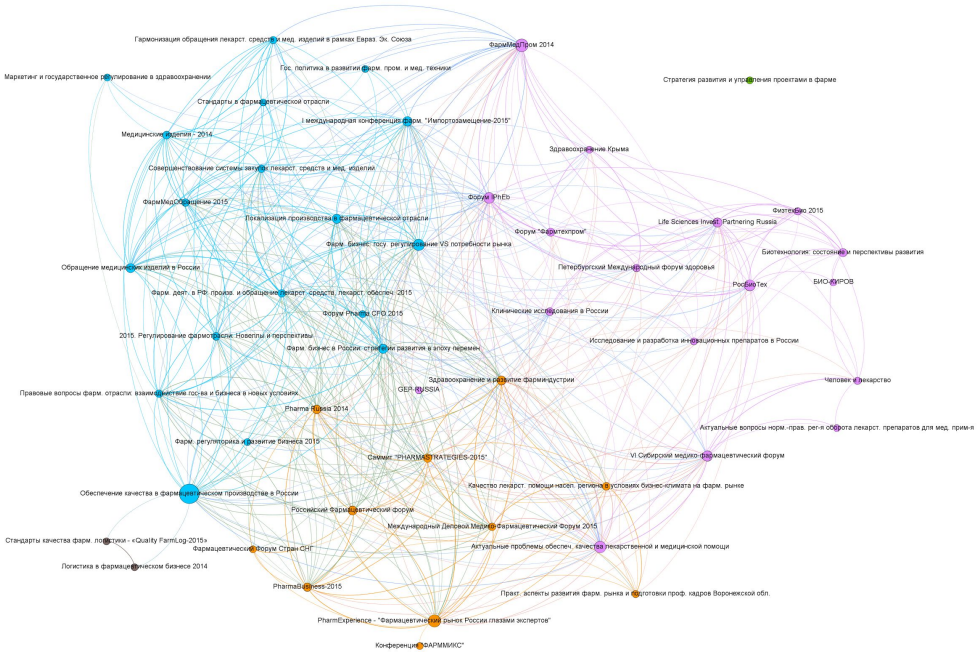
Коллокации

```
library(quanteda)

collocations <-
textstat_collocations(text, size = 2:3)
```

	collocation	count	length	lambda	z
1	главный герой	51	2	5.594665	26.268972
2	друг друг	36	2	5.457287	23.975265
3	самый дело	30	2	4.817666	20.631267
9	очень понравиться	21	2	3.518055	14.267273
4	главный героиня	19	2	6.262405	18.490280
8	год назад	15	2	6.308578	14.505164
5	сей пора	13	2	7.027718	16.648241

https://quickshout.blogspot.ru/2011/11/on-surface-of-it-just-word-looks-just.html

LSA - семантическая близость слов



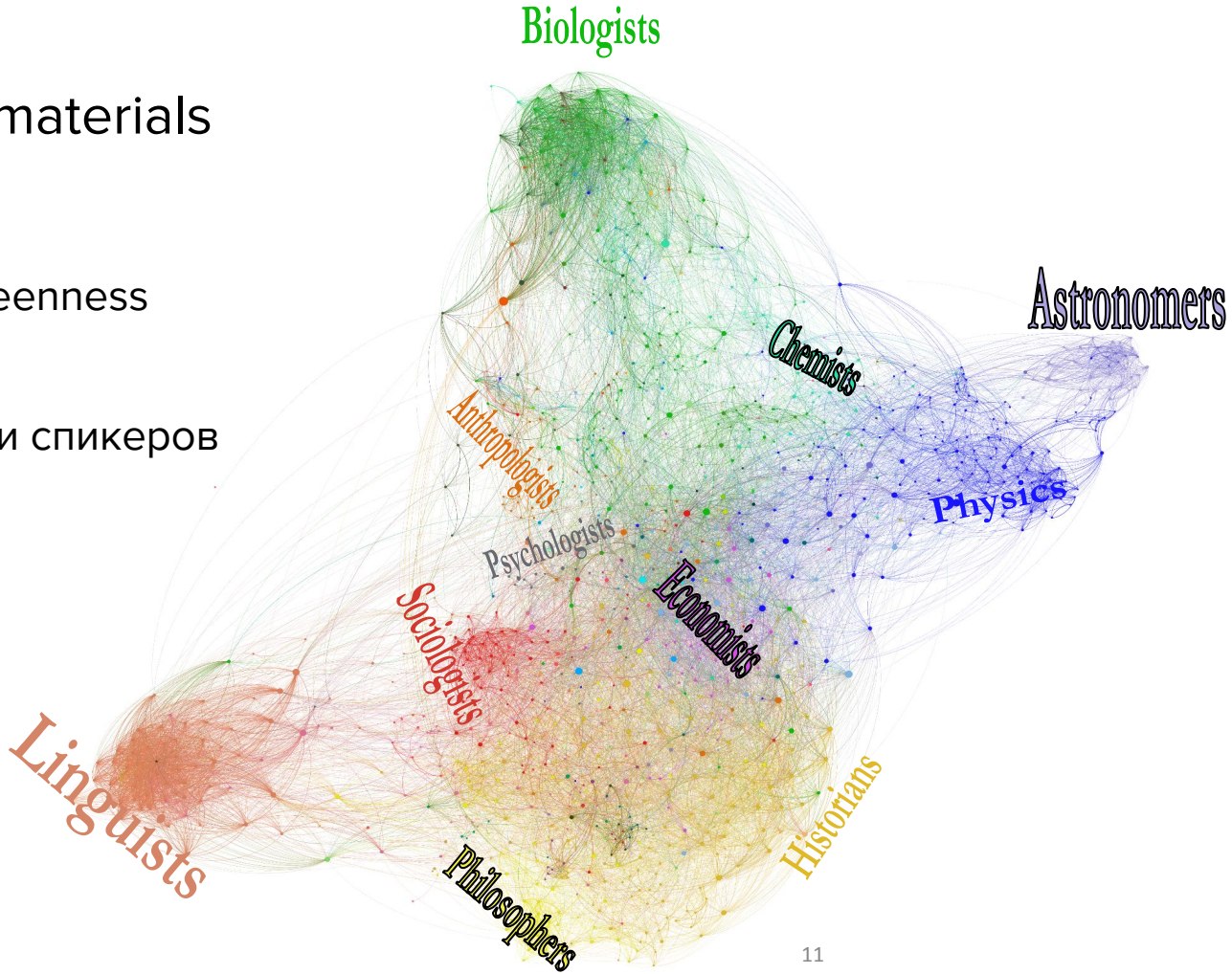

```
library(lsa)
tdm<-as.TermDocumentMatrix(dtmw)
lsa_space<-lsa(tdm, dims=dimcalc_share())
lsa_word_space<-lsa(dtmw, dims=dimcalc_share())

tdm_lsa<-as.textmatrix(lsa_space)
tdm_word_lsa<-as.textmatrix(lsa_word_space)

tdm_lsa[1:5,1:5] # что присвоено "tdm_lsa" (какое-то значение)
as.matrix(tdm)[1:5,1:5]
t.locs<-lsa_space$tk %*% diag(lsa_space$sk)
plot(t.locs,type="n")
text(t.locs, labels=rownames(lsa_space$tk))

lsa_space2<-lsa(tdm, dims=2)
t2.locs<-lsa_space2$tk %*% diag(lsa_space2$sk)
plot(t2.locs,type="n")
text(t2.locs, labels=rownames(lsa_space2$tk))

lsa.distances<-cosine(tdm_lsa) # косинусное расстояние между текстами в
LSA-пространстве
rownames(lsa.distances) <- farm$Name
colnames(lsa.distances) <- farm$Name
lsa.distances[upper.tri(lsa.distances)] <- NA
diag(lsa.distances)=NA
lsa.matrix <- melt(lsa.distances)
colnames(lsa.matrix) <- c("Source","Target", "Weight")
lsa.matrix<-lsa.matrix[lsa.matrix$Weight > 0, ]
lsa.matrix<-lsa.matrix[!(is.na(lsa.matrix$Weight)), ]

lsa.matrix$Weight2 <- lsa.matrix$Weight
lsa.matrix$Type <- "Undirected"
write.csv(lsa.matrix, "graph_farmo_lsa.csv")
View(order(lsa.matrix$Weight))

write.csv(labels_farma, "labels_farma.csv")
```

LSA - PostNauka materials

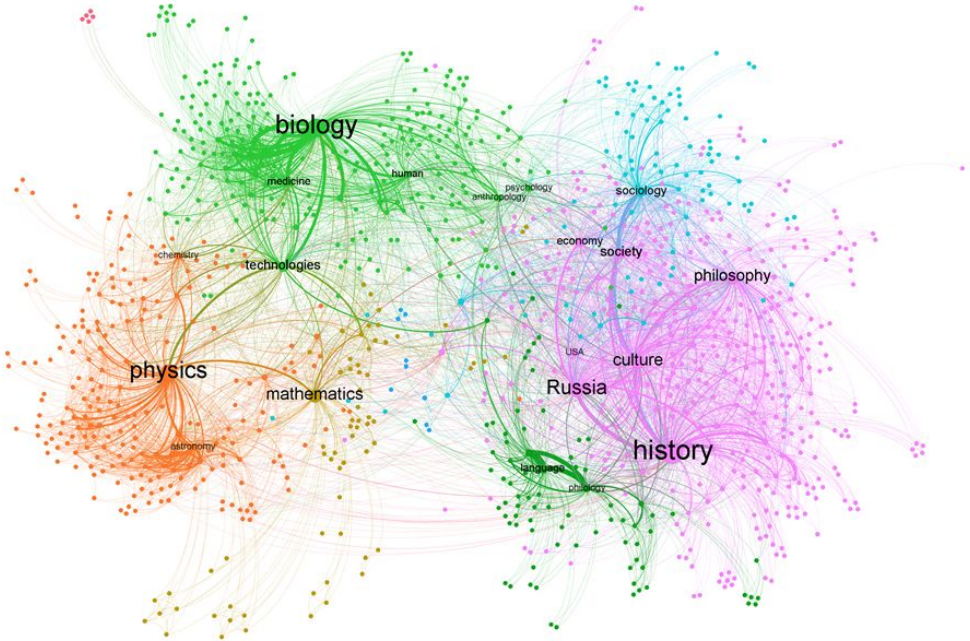
Размер узлов - Betweenness Centrality

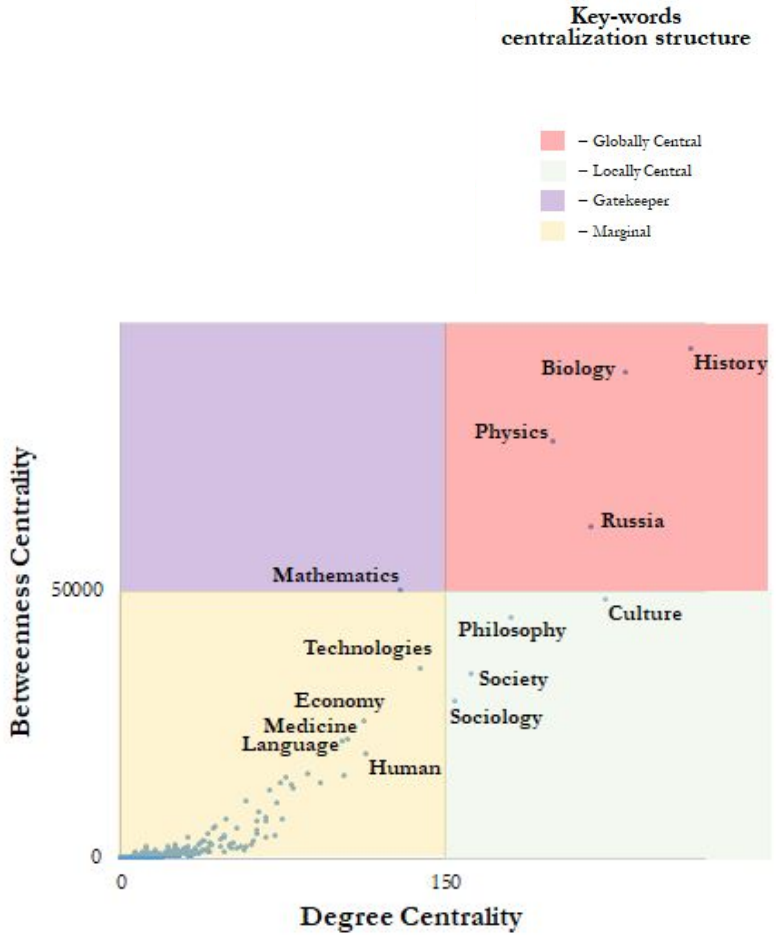
Цвет - специализации спикеров

Связь - LSA метрики



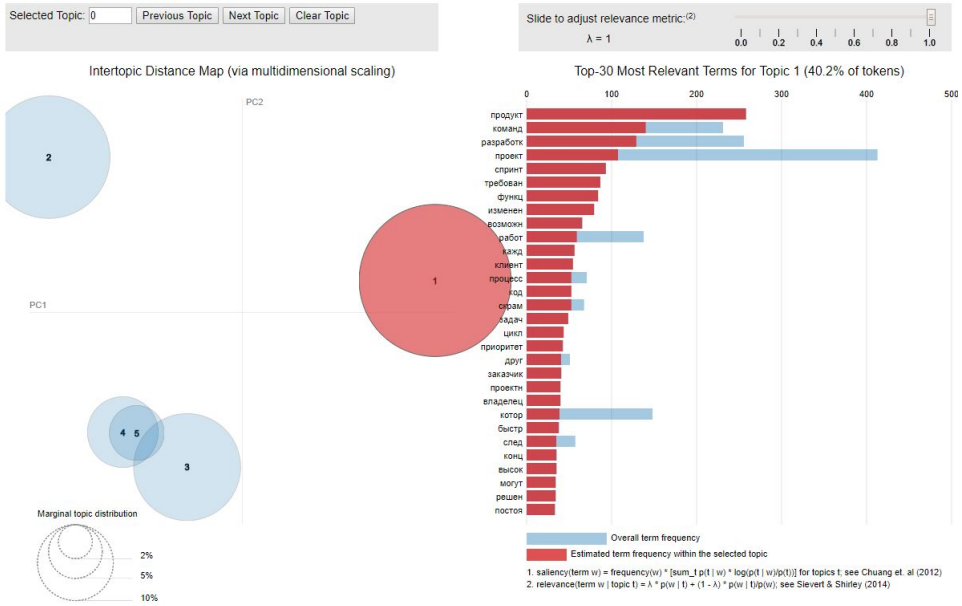
LSA - PostNauka materials





LDA — к каким темам относится документ

```
library(mallet)
mallet.instances <- mallet.import(id.array = tb$Название.раздела, text.array = corp, stoplist.file = "./data/stopwords.txt")
## настраиваем параметры модели и загружаем данные
topic.model <- MalletLDA(num.topics=5) # количество тем
topic.model$loadDocuments(mallet.instances)
topic.model$setAlphaOptimization(20, 50) # оптимизация гиперпараметров
## собираем статистику: словарь и частотность
vocabulary <- topic.model$getVocabulary() # словарь корпуса
word.freqs <- mallet.word.freqs(topic.model) # таблица частотности слов
## вершина частотного списка (по документной частоте)
head(word.freqs[order(word.freqs$doc.freq, decreasing=T),],30)
## параметр — количество итераций
topic.model$train(1000)
## выбор наилучшей темы для каждого токена
topic.model$maximize(10)
### LDA: выгрузка результатов
## таблица распределения тем по документам
doc.topics <- mallet.doc.topics(topic.model, smoothed=TRUE, normalized=TRUE)
## таблица распределения слов по темам
topic.words <- mallet.topic.words(topic.model, smoothed=TRUE, normalized=TRUE)
## метки для тем (по трем главным словам)
topic.labels <- mallet.topic.labels(topic.model, topic.words, 5)
```



[Ссылка на интерактивный граф]

LDA: этнические группы в ВК